Command Reference

Version 2016

PZFlex®

Thornton Tomasetti Inc.
10220 Stelling Rd, Ste. 100
Cupertino, CA 95014–2530
http://www.pzflex.com

CONTENTS

1 USING SYMBOL 1-1

PRINT CURRENT VALUE OF A SYMBOL VARIABLE 1-3
SYMBOL EVALUATION RULES 1-3
Changing Sign of Numerical Variables 1-4
Controlling Formats for Variable Substitutions in Text Strings 1-4
Inline Summation for Integers 1-5
THE **SYMB** AND **TEXT** STATEMENTS 1-6
SPECIAL OPTIONS OF THE **SYMB** STATEMENT 1-8
The #GET Keyword 1-9
The #SET Option 1-36
The #ADD Option 1-36
The #MSG Option 1-37
The #LIB Option 1-39
The #TITLE Option 1-39
The #SAVE Option 1-39
The #READ Option 1-40
The #LIST Option 1-40
The #MODIFY Option 1-41
The #REMOVE Option 1-41
The #ECHO Option 1-41
The #IGNORE Option 1-41
The #RAISE Option 1-42
The #PRINT Option 1-42
The #KILL Option 1-42
The #EXIT Option 1-42
The #DELFILE Option 1-42
The #SLEEP Option 1-43
The #SUBMIT Option 1-43
The #TEMP Option 1-43
The #TOOL Option 1-43
The #KEY Options 1-43
THE **IF**, **ELSEIF**, **ELSE** AND **ENDIF** STATEMENTS 1-45
THE **END$** STATEMENT 1-46
THE **PROC** (PROCEDURE) STATEMENT 1-47
The SAVE Option 1-47
The ERAS (Erase) Option 1-47
Executing a Procedure 1-48
Examples 1-49
THE **DO** STATEMENT 1-50
THE **GOTO** STATEMENT 1-52

2 COMMAND DEFINITIONS 2-1

DESCRIPTION OF COMMAND DEFINITIONS 2-1
COMMENT LINES 2-1
COMMAND STRUCTURE 2-2
Batch Option Command Structure 2-2
Batch Option Example 2-2
Interactive Option Command Structure 2-3
Interactive Option Example 2-3
FREE FORMAT SYNTAX 2-4
LIST OF **PZFLEX** COMMANDS 2-5
AXIS (axis) Input Command 2-19
BOND Input Command 2-28
BOUN (boundary) Input Command 2-33
CALC (calculate) Input Command 2-47

CIRC (circuit) Input Command....2-91
DATA Input Command....2-100
DRLX (dynamic relaxation) Input Command....2-139
DRLX (dynamic relaxation) Input Command....2-141
ECHO Input Command....2-142
EXEC (execute) Input Command....2-147
EXTR (extrapolation) Input Command....2-149
FFLD (freefield) Input Command....2-155
FLOW Input Command....2-163
FUNC (function) Input Command....2-165
GAGE Input Command....2-177
GEOM (geometry) Input Command....2-183
GLUE Input Command....2-195
GRID Input Command....2-199
GRPH (graphics) Input Command....2-200
GSMU (Grid Smoothing) Input Command....2-258
HEAT Input Command....2-263
JOB Input Command....2-268
LDEF (large deformation) Input Command....2-269
LINE (line element) Input Command....2-270
MAGN (magnetostrictive) Input Command....2-299
MASS Input Command....2-312
MATR (material) Input Command....2-317
MBRN (membrane) Input Command....2-382
MEM (memory) Input Command....2-387
MGR (manager) Input Command....2-388
MODL (Model) Input Command....2-389
MODS (modifications) Input Command....2-400
MP (multiprocessing) Input Command....2-413
OBJT Input Command....2-415
PIEZ (piezoelectric) Input Command....2-419
PLOD (pressure load) Input Command....2-449
POUT (printed output) Input Command....2-470
PRCS (process) Input Command....2-480
PRNT (print) Input Command....2-481
REGRID Input Command....2-482
REST (restart) Input Command....2-485
RIGD (rigid) Input Command....2-487
SET Input Command....2-500
SHAP (shape) Input Command....2-501
SHEL (shell element) Input Command....2-509
SHOW Input Command....2-533
SITE Input Command....2-534
STOP Input Command....2-559
TERM (terminal) Input Command....2-560
TIME Input Command....2-561
TITL (title) Input Command....2-563
USER Input Command....2-564
WNDO (window) Input Command....2-565
XFIL (X-files) Input Command....2-566
ZONE Input Command....2-571

3 REVIEW - A POSTPROCESSOR 3-1

COMMAND STRUCTURE 3-1
COMMAND DEFINITIONS 3-1
LIST OF **REVIEW** COMMANDS 3-2
ASOURCE (Acoustic Source) Input Command 3-7
AXIS Input Command 3-16
CONVERT Input Command 3-17
DATA Input Command 3-19
DERV (Derivative) Input Command 3-21
EXTR (Extrapolate - Kirchhoff) Input Command 3-22
FILT (Filter) Input Command 3-35
FREQ (frequency) Input Command 3-37
GRPH (graphics) Input Command 3-56
HIST (History) Input Command 3-76
ID (identifier) Input Command 3-82
INTG (Integrate) Input Command 3-83
LIST Input Command 3-84
MAKE Input Command 3-86
MEM (memory) Input Command 3-105
PS (PostScript) Input Command 3-107
READ Input Command 3-111
SCAL (scale) Input Command 3-113
SHFT (shift) Input Command 3-114
STOP Input Command 3-115
TERM (terminal) Input Command 3-116
WRIT (write) Input Command 3-117

4 BUILD - A GRID GENERATOR 4-1

LIST OF **BUILD** COMMANDS 4-2
AXIS Input Command 4-4
CORD (coordinate) Input Command 4-5
COPY Input Command 4-6
EQUV (equivalence) Input Command 4-8
FILL Input Command 4-9
GEOM (geometry) Input Command 4-11
GRID Input Command 4-16
LOCK Input Command 4-17
LVAL (line value) Input Command 4-18
PRNT (print) Input Command 4-19
QUAD (quad region) Input Command 4-20
HEX (HEX region) Input Command 4-21
RADI (Radius) Input Command 4-22
READ Input Command 4-23
RFIL (repetitive fill) Input Command 4-24
ROT (rotate) Input Command 4-26
RPOS (relative position) Input Command 4-27
SCAL (scale) Input Command 4-28
SET Input Command 4-29
SHFT (shift) Input Command 4-30
SOR (solid/surface of revolution) Input Command 4-31
STOP Input Command 4-32
SVAL (surface value) Input Command 4-33
SWTH (switch) Input Command 4-35
TERM (terminal) Input Command 4-36
TITL (title) Input Command 4-37
WRIT (write) Input Command 4-38

1 USING SYMBOL

Symbol is an interpretive programming and job control language that is an integral part of SOFSHEL, the software support shell developed by Thornton Tomasetti to facilitate software development, integration, and usability. Symbol is resident in any application, such as **PZFlex**, **Build,** or **Review**, that has been developed within the SOFSHEL architecture.

A new user need not know anything about Symbol to use the application. Experienced users, however, will find the capabilities afforded to them by Symbol to be very useful. Symbol allows the user to place variable names as arguments on any command line, to embed programming statements (variable assignments, mathematical operations, looping and branching) within the Job Input File and to define procedures (macros) consisting of any collection of program commands that can be invoked with a single command. Taken together, these capabilities afford the user flexibility in setting up input command files and allow the development of self-adaptive input files termed "templates." A template is an input command file that uses variables instead of explicit input values and uses of arithmetic relations to define other values, which are functions of key input parameters. Key input parameters can be edited into the input file, read in from an external file, or interactively provided by the user.

The Symbol language is composed of the following statement types, which may be placed wherever primary or subcommands occur within the input command file: **SYMB**, **TEXT**, **GOTO**, **END$**, **DO**, **PROC, IF, ELSEIF, ELSE, ENDIF**. They may start in column 1 like a primary command or in a succeeding column without affecting the processing of other primary and subcommands. The functionality provided by the different statement types are:

1 The **SYMB** (Symbol) statement provides the capability to assign values to symbol variables. Once assigned, these variables may be used anywhere in the input stream of a job by preceding the variable name with a $ character. Variables are either numeric (integer or real) or character variables up to 8 characters long. The values of variables may be defined explicitly, with a mathematical expression, accessed from within the program's own internal variable tables, or provided interactively by the user. A number of other functions related to handling symbol variables (saving symbols to a file, reading them from a file, etc.) and controlling jobs are also provided with the **SYMB** statement.

> The **SYMBX** statement is like **SYMB** but once set, cannot be changed. This command is used when setting variables of interest in the batch controller.

2 The **TEXT** statement provides the capability to assign a text string of up to 80 characters as the value of a symbol variable.

3. The **GOTO** and **END$** statements provide the ability to branch to any point in the input stream of a job. Alternatively, the **IF**, **ELSEIF**, **ELSE,** and **ENDIF** statements provide structured, conditional branching constructs.

4 The **DO** statement provides the equivalent of Fortran do loops, i.e., automatic looping within the input command stream.

5 The **PROC** (Procedure) statement provides the functionality to define and execute procedures. Procedures are user-defined groups of primary and/or subcommands (i.e., a macro) that the user may use more than once. Once defined, the entire group of commands is processed each time the procedure is invoked.

Example 6-1 illustrates the use of the **SYMB** statement to define the values of several variables and the use of these variables as input parameters on other command lines. This is a portion of a template designed to Build automatically a **PZFlex** model given the spatial extent of the grid, the material constants of the continuum, and the highest frequency desired to be properly represented by the model.

```
c       Use Symbol to determine the discretization required for the model

c       Specify the spatial extent of the grid

symb    xgrid = 500.
symb    ygrid = 300.

c       Specify the highest frequency needed to be well-resolved by the grid

symb    freqhigh = 10

c       Specify the density and dilatational and shear wave speeds of the continuum

symb    density = 50.
symb    cp = 6.
symb    cs = 4.

c       Compute the parameters required to define the model

c       Period and wavelength of highest frequency (shortest wavelength) in model

symb    period = 1. / $freqhigh
symb    wavelng = $period * $cs

c       Element size required to properly resolve this wavelength
symb    esize = $wavelng / 12.

c       Nearest integer number of elements in the x and y directions

symb    nx = 1 + nint ( $xgrid / $esize )
symb    ny = 1 + nint ( $ygrid / $esize )

c       Now define the grid using standard PZFlex commands with variable input parameters

grid    $nx $ny
geom
        xcrd        0.0 $xgrid
        ycrd        0.0 $ygrid
        end
matr
        type        elas
        wvsp        on
        prop        rock $density $cp $cs
        end
site    regn        rock

C       (etc.)
```

Example 6-1. Partial Template Illustrating Use of the Symbol Language

PRINT CURRENT VALUE OF A SYMBOL VARIABLE

In interactive mode, it is often of interest to know the value of a symbol variable. The simplest way to display the current value of a Symbol variable is simply to use the following syntax:

SYMB vname

where vname is the name of the variable. For example:

symb n1

prints the value of N1 to the current output file (or the terminal screen when in interactive mode).

SYMBOL EVALUATION RULES

Once defined, a symbol variable may be used on any succeeding command line. A symbol variable is identified to the input interpreter by preceding the variable name with a $ sign. The input interpreter identifies the variable name and substitutes the value of the variable in its place. Standard syntax rules require that the variable name must be delimited by blanks and cannot contain embedded blanks. The following example illustrates this point. The values of variable N1 and N2 are defined and then used to compute the value of N3.

```
symb n1 = 20
symb n2 = 40
symb n3 = $n1 + $n2
```

The input interpreter performs variable substitution in a single pass from right to left as shown here.

SYMB N4 = $N1 + $N2 + $N3 <-- Parser starts from here and works to the left side

This approach allows for variable names to be composed of other variables as shown below. Several examples of the symbol substitution functionality are:

```
symb n = 3
symb sum3 = 300.7
symb name = sum3
symb test1 = $sum$n
symb test2 = $$name
```

where, for the last assignment statement, the interpreter first evaluates $NAME to SUM3 before proceeding to evaluate $SUM3 and consequently, the value of TEST2 will be 300.7. Similarly, TEST1 will also be set to 300.7 since $N will first be evaluated to 3 before proceeding to evaluate $SUM3.

In certain situations, the user may need to guide the input interpreter by using parenthesis to physically denote the name of Symbol variables to evaluate. For example:

```
symb nrun = 3
symb name = A
text file = model${name}${nrun}results
```

This sets text variable FILE to MODELA3RESULTS. If the parentheses are not included, the interpreter does not evaluate the variable names properly.

The input interpreter identifies a variable name to evaluate as the string between the $ and a terminating character. Terminating characters include: blank, period, comma, $, (,) , and /. Therefore, the following text string representing a file name would be set to /USA/RUN1.20

```
symb job = 1
symb project = usa
symb index = 20
text filename = /$project/run$job.$index
```

Changing Sign of Numerical Variables

A sign character (+ or -) may precede the $ sign of a numeric symbol variable. If the minus sign is used (i.e., -$WEIGHT), the negative of the symbol value is used.

Controlling Formats for Variable Substitutions in Text Strings

When arithmetic operations are performed, the full numerical precision of all variables is maintained throughout the Symbol substitution process. However, when performing symbol substitution within text strings to generate labels, filenames, etc., a user may need control over the formatting of the text substitution. Typically, a user will want to control the appearance of the values of floating point variables. For example,

```
symb theta = 90.5
symb phi = 25.5
test filename = DATA.THETA.$theta.PHI.$phi
```

if a user generates a file name containing the value of real variables THETA and PHI, the default format for a floating point value substitution is (g15.7) in Fortran convention which produces the following value for FILENAME:

DATA.THETA.90.50000.PHI.25.50000

To change the format, a user must define the format of the variable of interest using Fortran format syntax for a single variable. The symbol name must be followed immediately by the % sign and then a valid single variable format enclosed in parentheses The following illustrates the use of the formatting construct.

```
text filename = DATA.THETA.$theta.PHI.$phi%(F5.1)
```

Once a format for a particular variable type is specified, the default format for that type of variable is set to the format declaration and used in interpreting the rest of the input line. As the input interpreter evaluates the line from right to left, declaring a format for PHI causes the same format to be used for THETA, producing the following value for FILENAME:

DATA.THETA.90.5.PHI.25.5

Note that leading blanks produced by a specified format are stripped by default during the substitution process. Succeeding input lines revert to the original default format for real variable substitutions.

In certain cases, a user may not want the leading blanks stripped during the substitution process. Embedding the "<" character between the % and the start of the format specification causes any leading blanks to be present in the substituted string. Embedding the ">" character between the % and the start of the format specification causes any leading blanks to be filled with zeros in the substituted string. For example, the following statements:

```
symb n1 = 20
symb data1 = 30.5238
text LABEL1 = 'VALUES:$n1 & $data1'
text LABEL2 = 'VALUES:$n1%(I5) & $data1%(F6.1)'
text LABEL3 = 'VALUES:$n1%<(I5) & $data1%<(F6.1)'
text LABEL4 = 'VALUES:$n1%>(I5) & $data1%>(F6.1)'
```

produce the following results:

```
LABEL1      =VALUES:20 & 30.52380
LABEL2      =VALUES:20 & 30.5
LABEL3      =VALUES:  20 &   30.5
LABEL4      =VALUES:00020 & 0030.5
```

Inline Summation for Integers

Integer variables and constants may be concatenated with the + and - operators to produce an inline expression (with no embedded blanks) that is evaluated automatically at the time it is encountered. The following example shows two equivalent ways of defining the number of nodes on a GRID command. The first specifies the number of nodes explicitly. The second specifies the number of elements explicitly and uses inline integer summation to add 1 to each value to produce the actual number of nodes.

```
symb      inode = 10
symb      jnode = 20
grid      $inode $jnode        /* the number of nodes is explicitly declared

symb      ielem = 9
symb      jelem = 19
grid      $ielem+1 $jelem+1    /* the number of elements is incremented by 1
```

THE **SYMB** and **TEXT** STATEMENTS

The **SYMB** (Symbol) statement is used primarily to assign a value to a variable name; it may be up to 8 characters long. Variables are typed as integer numeric, real numeric, or character. The value of a numeric variable can be assigned by setting it equal to a number or to a mathematical expression. Implicit typing of numeric variables as either integer or real is done as in Fortran; all variables are assumed real unless they begin with the letters I through N, in which case they are assumed to be integer. Mathematical expressions are similar to Fortran expressions. Blank delimiters, however, are required between all items (number, variable, and mathematical operator) in the expression.

Character strings of up to 8 characters may be assigned with the **SYMB** command. For longer strings, the **TEXT** command can be used to assign character strings of up to 80 characters to a variable name. A character variable may begin with any letter.

The basic **SYMB** statement is:

SYMB vname = value

where:

vname = the variable name
value = the value of the variable

The variable *vname* must begin with an alphabetic character and can be from 1 to 8 characters long. The variable denoted by *vname* may be real, integer, or character data. Variables that begin with the letters I through N, and that are assigned a numeric value are implicitly typed as integer. Variables beginning with any other letter, and which are assigned a numeric value, are implicitly typed as real. Any variable that is set equal to a character data field is implicitly typed as a character variable. The **SYMB** statement can be used to assign a character string of up to 8 characters to a variable name. Longer strings will be truncated to the first 8 characters of the string. The **TEXT** command is used to assign longer text strings to a variable and has the form:

TEXT vname = string

where *string* is any character string. If *string* includes embedded blanks, it must be enclosed in single quotes. The TEXT command is more limited than SYMB. It supports only the IF EXIST/ NOEXIST conditional described below. Examples of the use of **SYMB** and **TEXT** commands to assign character variables are:

```
symb speed = 38.756                              /* real number assignment
symb index = 12                                  /* integer assignment
symb option = on                                 /* short character string assignment
text filename = /USR/FILES/DATA                  /* long character string assignment
text title = 'FIGURE 1. CALCULATION RESULTS'     /* with embedded blanks
```

The **SYMB** statement can also take the form:

SYMB *vname* = <expression>

The <expression> may be composed of the following mathematical operations: COS, SIN, TAN, EXP, ALOG, ALOG10, ACOS, ASIN, ATAN, ATAN2, SQRT, ABS, SIGN, INT, NINT, MAX, MIN, **, *, /, +, and -.

Parentheses may be used to control the sequence in which the expression is evaluated (innermost parentheses evaluated first). Within parentheses, the evaluation is done from left to right for each operator listed above in the order listed. The real number equivalence of all arguments to COS, SIN, TAN, EXP, ALOG, ALOG10, ACOS, ASIN, ATAN, ATAN2, SQRT, ABS, INT, and NINT are used. Note that Symbol does not allow a real expression in which a negative number is taken to an exponential power (even the power of 2), as the real number representation of the exponent is used in the expression and math libraries do not allow this operation. Multiple arguments may be provided to the MAX, MIN, ATAN2, and SIGN functions. These arguments must be separated by commas (which are themselves blank delimited) and be bounded by parentheses.

Examples of arithmetic expressions are:

```
symb x = sqrt ( 5. ** 2 + 10. ** 2 )          /* compute hypotenuse of triangle
symb maxm = max ( $m1 , $m2 , $m3 )           /* compute maximum of 3 values
symb pi = 4. * atan ( 1. )                    /* compute value of pi
```

The two forms of the **SYMB** statement discussed above can also have conditional operators appended to them. The **SYMB** statement with a conditional operator has the form:

SYMB vname = value IF datum1 op datum2

SYMB *vname* = <expression> IF *datum1 op datum2*

where *op* is any of the conditional operators: EQ, NE, LT, LE, GT, and GE whose meanings correspond to their usage in Fortran. *datum1* and *datum2* can be either numeric or character data. If the values being compared are numeric, they are compared as real numbers (i.e., the real number representation of an integer is used in the comparison). The effect of the conditional operator is: (1) If the conditional is true, then the symbol value assignment is made. (2) If the conditional is not true, then this **SYMB** statement is skipped and the assignment is not made. One other type of conditional assignment is based on the current existence of a variable name. The conditional can take the following two forms:

SYMB *vname* = *value* IF EXIST

SYMB *vname* = *value* IF NOEXIST

where EXIST and NOEXIST refer to the variable name. In the first case, if *vname* currently exists, the conditional is true and the value assignment is made. If *vname* does not exist, no assignment is made. The second case is the opposite. Multiple conditional operators can be appended to an expression. This type of expression takes the form:

SYMB *vname* = <expression> IF *datum1 op datum2* IF *datum3 op datum4* etc.

For a symbol equivalence with multiple conditionals, all conditionals must be true before the value assignment is made.

SPECIAL OPTIONS OF THE **SYMB** STATEMENT

The **SYMB** statement has a number of other options besides those discussed above. All the special options of the **SYMB** statement are activated by a keyword preceded by the # sign as follows:

SYMB *#keyword parameter1 ... parametern*

The keyword options and their functions are:

1. #GET - assigns the value of a variable based on internal information within the executing program.
2. #SET - assigns a value to an internal Data Manager array location.
3. #MSG - writes a message to the terminal screen or to a file. Also can be used to receive variable assignment information from the keyboard.
4. #TITLE — requests that the user provide titling information for the current job.
5. #SAVE - saves the current symbol variables to an external file.
6 #READ - reads a set of symbol variables or any other group of input commands from an external file.
7. #LIST - lists the values of a group of symbol variables.
8 #MODIFY - modifies the values of a subgroup of symbol variables.
9. #REMOVE - removes a symbol name from the current symbol tables.
10 #ECHO - allows the user to control the amount of output generated by **SYMB** statements.
11. #PRINT - causes all following symbol names printed to the output file to be expanded with their present value.
12. #KILL - causes the program to check for the presence of a job kill file.
13. #EXIT - terminates the program.
14. #DELFILE - deletes a disk file.
15. #SLEEP - causes the program to sleep for a specified length of time.
16. #SUBMIT - submits a process to the system.
17. #TOOL - invokes a standard tool.
18. #TEMP - invokes a standard template.
19. #LIB - reads a group of input commands from a dated external file.
20. #KEY — helps generate a set of keypoints and copy out locations for simplified model building.

The purpose, form, and parameters of each keyword are discussed.

The #GET Keyword

The #GET keyword allows one or more variables to be assigned values by the program based on the function specified. The value assigned may be a data array value stored within the Data Manager or internal code information such as run time, model time step, etc. The #GET option has the form:

SYMB #GET { vname1 vname2 ... vnamen } function [parameters]

where:

vname1, vname2 etc = the names of symbol variables which will be assigned values by this command. The number of values returned depends on *function*. Braces,{ }, bound the variable names. If only a single variable is assigned, the braces may be omitted.

function = the function that defines what values are to be assigned by this command.

parameters = input parameters required by *function*. The number and type of parameters vary depending on which function is being used.

The user need not specify variable names for all values that will be returned by the #GET function. For example, if a user uses the CRDNODE function, which returns the x-, y-, and z-coordinates of a node, but needs only the y-coordinate, it is necessary to specify only two variable names between the braces. The z-coordinate will not be assigned to any variable.

Examples of the use of the #GET keyword option include:

Assigns the current cpu time to variable VALU:

symb #get { valu } cputime

Assigns the nodal indices of the node closest to coordinate x=3., y=4. and z=5. to variables I, j, and k.

symb #get { i j k } clsnode 3. 4. 5.

Assigns the value of Data Manager array DAT1(3,9,4) to variable VALUE.

symb #get { value } array DAT1 3 9 4

Assigns information about the maximum y point of the eighth curve on local file F1 for the first 1000 sampling points.

symb #get { xmax ymax imax } curvmax F1 8 1 1000

Assigns information about the minimum y point of the eighth curve on local file F1 for the x-data range of the curve (1.0e6 -> 10.e6)

symb #get { xmax ymax imax } curvmin F1 8 1.0e6 10.0e6

Computes the width of the signal which is above 20 db of the peak signal strength. Search from the peak value out to the ends of the entire record.

symb #get { width xl xr status } curvwndo F1 8 20. db out

For easier reference, the #GET commands appear in table format over the next few pages. More detailed explanations of the commands can be found in the relevant sections.

Tables of #GET Commands

Job Information

Function Name	Description
autostatic	Returns the status of the convergence if DRLX being used
codename	Returns the code type running (PZFlex, Review, etc)
coderev	Returns code version
convention	Returns suffix or prefix file naming convention
datetime	Returns the current date and time as text
envar	Returns system environment variables
findfile	Returns “found” or “notfound” for search file specified
ident	Returns the job identifier
jobname	Returns the name of the job, i.e., <jobname>.flxinp
precision	Returns the current floating point precision
ncpu	Returns number of cpu sockets
ncore	Returns number of cores per socket
ncycles	Returns the maximum subcycling level in any zone
sysname	Returns computer operating system name
timestep	Returns the current timestep size

Time information

Function Name	Description
cputime	Returns the current CPU time for a job
itimdata	Returns the timestep number associated with the last data array
itimstep	Returns the current simulation time step number
iwallclock	Returns the current wallclock time of an ongoing simulation
timedata	Returns the time associated with the last data array imported
timenow	Returns current simulation time of a calculation
timestep	Returns the current timestep
wtime	Returns the current wall clock time in seconds

Data Manager

Function Name	Description
array	Returns the value from a named array
datamin	Returns the minimum value of a named array
datamax	Returns the maximum value of a named array
datasum	Returns the sum and average value of a named array
findarray	Returns “found” or “notfound” for search array specified
histval	Finds the magnitude value for a time.frequency set, if possible
mgrinfo	Returns the data manager information for a specifed data array
volmintg	Returns the volume integral of data values in an array
levelvol	Returns the volume in model

Coordinate Data

Function Name	Description
clselem	Returns the indices of the closest element to the given location
clshex	Returns the indiceis of the closest hex element to given location
clsnode	Returns the indices of the closest node to the given location
clstet	Returns the indicies of the closest tet element to the given location
clsplod	Returns the face of the closest pressure load to a given location
clsshel	Returns the indices for the closest shell to a given location
crdelem	Returns the coordinates of the specified element
crdglob	Returns the coordinates of local space axis from global location
crdlocl	Returns the coordinates of global space axis from local location
crdnode	Returns the coordinates of the specified node
crdtet	Returns the coordinates of the specified tet
distelem	Returns the distance between 2 element indices locations
distnode	Returns the distance between 2 node indices locations
gconloc	Returns the global indices location for a given GCON node
indxloc	Returns the global ijk indices for a given global index number
locindx	Returns the global index number for a given set of ijk indices
matbound	Returns the spatial coordinates or indices for a named material

Elemental Data

Function Name	Description
count	Returns the total number of objects of type *type* in the model
clselem	Returns the indices of the closest element to the given location
crdelem	Returns the coordinates of the specified element
crdtet	Returns the coordinates of the specified tet element
distelem	Returns the distance between 2 element indices locations
elemnum	Returns the element number corresponding to label *id*
matelem	Returns the name of the material for a given ijk-indices
modelem	Returns the total number of elements eroded
label	Returns the label *id* corresponding to element number *ielm*

Nodal Data

Function Name	Description
clsnode	Returns the indices of the closest node to the given location
crdnode	Returns the coordinates of the specified node
delpath	Returns the path length along a row of nodes
distnode	Returns the distance between 2 node indices locations

Material Information

Function Name	Description
matbound	Returns the spatial coordinates or indices for a named material
matelem	Returns the material name for a given ijk-indices
matprop	Returns the material value number for a given material name
matnum	Returns the material number for material name
mattet	Returns the material name for the given tet element

Graphical Information

Function Name	Description
movie	Returns number of frames in current animation
view	Returns the values for eye, view, and vpnt for the given plot
window	Returns the spatial coordinates of the last plot
modelbounds	Returns the bounding coordinates of the physical model

Mods Information

Function Name	Description
modelem	Returns the total number of elements eroded
modvol	Returns the total volume of elements eroded

Load Information

Function Name	Description
airblast	Returns data relating to the air pressures from a blast (CWEP)
clsplod	Returns the face of the closest pressure load to the given location
pmapblast	Returns data relating to the air pressures from a blast (PMAP)

Character Information

Function Name	Description
charlen	Returns the length of the given character string
charloc	Returns the location of the substring within the given string
charset	Inserts a new character string in an existing one

Miscellaneous Information

Function name	Description
circuit	Returns the indices of the nodes for the named circuit
compwndo	Returns the current size of the computational window
echotime	Returns the difference between model time and actual time
findfile	Returns “found” or “notfound” for search file specified
ncycles	Returns the number of zones in the model
random	Creates and returns a random number
tablmatr	Returns number of materials on the site input file.
tablsiz	Returns coordinates and numer of cells on a site input file
term	Returns the current state of the terminal

Symbol Information

Function name	Description
symbtype	Returns the symbol type
rootmax	Returns largest integer suffix of a symbol beginning with *rootname*

Review Information

Function name	Description
curvcord	Returns the coordinates for the record in the local file
curvcros	Returns the x and y values for a given y value crossing
curvdelx	Returns the difference between steps on the x axis
curvijkl	Returns the ijk indices and label for the record in the local file
curvmax	Returns the maximum value on a given record
curvmaxs	Returns the x and y values for the local maximums on records
curvmin	Returns the minimum value on a given record
curvmins	Returns the x and y values for the local minimums on records
curvnval	Returns the number of x-y values in the local file
curvnrec	Returns the number of records in the local file
curvrecd	Returns the record number in the local file for a given location
curvtitl	Returns the title of the local file
curvtshift	Returns time shift information for a specific time history
curvtype	Returns the type of local file
curvvalu	Returns the x and y value for an abscissa value on a specific record
curvwndo	Returns values for curve that exceeds a given y value
curvxmax	Returns the maximum x value for the record
curvxmin	Returns the minimum x value for the record
snapmax	Returns the total number of snapshots in the local file

Job Information

AUTOSTATIC: when using the **DRLX** command and the STAT option on the **EXEC** command, this option returns the status of convergence to a user-specified tolerance. The convergence status is returned in variable *cstatus*. Possible values are NOTSET =no dynamic relaxation in progress, NO=convergence has not been achieved, YES=convergence was achieved. Variable rvalue contains the current convergence value (the ration of the current velocity norm to the maximum velocity norm that occurred during the analysis).

#GET { *cstatus rvalue* } AUTOSTATIC

CODENAME: returns the name of the code running this job (PZFlex, Review, or Build).

#GET { *cname* } CODENAME

CODEREV: returns the code version for this job.

#GET { *cname* } CODEREV

CONVENTION: returns the file naming convention suffix/prefix (e.g flxinp.job or job.flxinp) for this job.

#GET { *cname* } CONVENTION

DATETIME: returns a text string containing the current date and time.

#GET { *cdate* } DATETIME

ENVVAR: returns system environment variables. Valid *variablenames* include "home," "key," and "user" in addition to other machine-dependent environment variables. (Note: on some machines environment variables must be capitalized.)

#GET { *vname* } ENVVAR *variablename*

FINDFILE: returns whether or not the disk file *filename* exists. The result is returned in character variable *cfind*. *cfind* is set to "found" if the *filename* exists. If not, *cfind* is set to "notfound."

#GET { *cfind* } FINDFILE *dataname*

IDENT: returns the job identifier in character variable *cid*

#GET { *cid* } IDENT

Job Information (continued)

JOBNAME: returns the job name for this job.

#GET { *cname* } JOBNAME

PRECISION: returns the current floating point precision (single or double) in character variable *cprec*.

#GET { *cprec* } PRECISION

NCORE: returns the number of cores per CPU socket on the computer

#GET { *ncore* } NCORE

NCPU: returns the number of CPU sockets on the computer

#GET { *ncpu* } NCPU

NCYCLES: returns the maximum number of subcycling levels for all computational zones in integer variable incyc. (Available only after the **PRCS** step in PZFlex, EMFlex, etc.)

#GET { *incyc* } NCYCLES

SYSNAME: returns the operating system name for this job.

#GET { *cname* }SYSNAME

TIMESTEP: returns the current model time step used for the analysis in real variable *rstep*.(Available only after the **PRCS** step in PZFlex, EMFlex, etc.)

#GET { *rstep* } TIMESTEP

Time Information

CPUTIME: returns the current cpu time for a job in real variable *rcpu*.

#GET { *rcpu* } CPUTIME

ITIMDATA: returns the time step number associated with the last data array imported using the **DATA IN** command in integer variable *itime*.

#GET { *itime* } ITIMDATA

ITIMSTEP: returns the current simulation time step number of an ongoing calculation in integer variable *itime.* (Available only after the **PRCS** step in PZFlex, EMFlex, etc.)

#GET { *itime* } ITIMSTEP

IWALLCLOCK: returns the current wallclock time of an ongoing calculation in integer variables *numsectotal.* The breakdown of this total into, seconds, minutes, hours, and days is also provided in *numsec* (seconds), *nummin* (minutes), *numhr* (hours), and *numday* (days).

#GET { *numsectotal numsec nummin numhr numday* } IWALLCLOCK

TIMEDATA: returns the time associated with the last data array imported using the **DATA IN** command in real variable *rtime*.

#GET { *rtime* } TIMEDATA

TIMENOW: returns the current simulation time of an ongoing calculation in real variable *rtime*. (available only after the **PRCS** step in PZFlex, EMFlex, etc.)

#GET { *rtime* } TIMENOW

TIMESTEP: returns the current model time step used for the analysis in real variable *rstep*.(Available only after the **PRCS** step in PZFlex, EMFlex, etc.)

#GET { *rstep* } TIMESTEP

WTIME returns the current wallclock time in seconds to high precision in a real variablefor code compiled twith the Intel compiler.

#GET { *seconds* } WTIME

Data Manager

ARRAY: returns the value of *dataname*, a Data Manager array, having the array indices *i, j, and k* in variable *value.* The data type of variable *value* must be the same as the data type of the array.

#GET { *value* } ARRAY *dataname i j k*

DATAMIN: returns the minimum value of *dataname*, a Data Manager array (type real), in real variable *rvmin* and the indices of the value in integer variables *ii, ij,* and *ik.* If only *dataname* is provided, the entire array is searched. If an indices window is specified by *ibegin, iend,* etc., only values within the indices window are searched.

#GET { *rvmin ii ij ik* } DATAMIN *dataname ibegin iend jbegin jend kbegin kend*

DATAMAX: same as DATAMIN except the maximum value is returned in real variable *rvmax*.

#GET { *rvmax ii ij ik* } DATAMAX *dataname ibegin iend jbegin jend kbegin kend*

DATASUM: returns the sum and average of a group of data values in *dataname*, a Data Manager array (type real), in real variables *rsum* and *raverag,e* and the number of values summed in integer variable *invalue.* If only *dataname* is provided, the entire array is summed. If an indices window is specified by *ibegin, iend,* etc., only values within the indices window are summed.

#GET { *rsum raverage invalue* } DATASUM *dataname ibegin iend jbegin jend &*
kbegin kend

FINDARAY: returns whether or not the array *dataname* exists in the Data Manager. The result is returned in character variable *cfind. cfind* is set to "found" if *dataname* exist. If not, *cfind* is set to "notfound."

#GET { *cfind* } FINDARAY *dataname*

HISTVAL: linearly interpolates the 2-column array *histname* (typically created via the DATA HIST command), to compute *yval* corresponding to *xval.* x is the first column, and y is the second. If *xval* is outside the range of x, the first (last) value and a warning are returned.

#GET { *yval* } HISTVAL *histname xval*

MGRINFO: returns Data Manager information for array *dataname.* The number of dimensions of the array is returned in integer variable *indim,* and the range of each array indices is returned in integer variables *iirng, ijrng,* and *ikrng.* The type of data array is returned in character variable *ctype. ctype* will be "i" for integer arrays, "f" for real floating point arrays, "d" for double precision floating point arrays, and "c" for character data arrays.

#GET { *indim iirng ijrng ikrng ctype* } MGRINFO *dataname*

Data Manager (continued)

VOLMINTG: returns the volume integral of data values in Data Manager array *dataname* (type real), in real variable *volintg*. Also returns the element volume in *volume* and the number of values summed in integer variable *nvalue*. If only *dataname* is provided, the entire array is summed. If an indices window is specified by *ibegin*, *iend*, etc., only values within the indices window are summed. Only elements of type *matname* are integrated when *matname* is specified.

#GET { *volintg volume nvalue* } VOLMINTG *dataname ibegin iend jbegin jend & kbegin kend matname*

LEVELVOL: returns the volume in model x,y,z-space of data in an elemental or nodal data manager array that meets a given threshold. The minimum and maximum extents of the volume (ie. the bounding box) are also returned. *dataname* is the elemental or nodal array containing the data, *threshold* is the data value level — elements with data values meeting this threshold are included in the volume calculation (nodal data is first interpolated to the elements). *option* is one of GT, LT, GE, LE or EQ to indicate that data must be "greater than", "less than", "greater than or equal to", etc. the threshold to be included. *ibegin – kend* can be included to limit the calculation to these index bounds.

#GET { *volume xmin xmax ymin ymax zmin zmax* } LEVELVOL *dataname threshold option & ibegin iend jbegin jend kbegin kend*

Coordinate Data

CLSELEM: same as CLSNODE except the indices are for the closest continuum element and the distance is to the element's center. Entering SKPVOID will find the clossest non-void element

#GET { *ii ij ik rdist* } CLSELEM (*grid*) *x y z ibegin iend jbegin jend kbegin kend (SKPVOID)*

CLSHEX: same as CLSNODE except the indices are for the closest gcon hex element and the distance is to the element's center.

#GET { *ii rdist* } CLSHEX (*grid*) *x y z*

CLSTET: same as CLSNODE except the indices are for the closest gcon tet element and the distance is to the element's center.

#GET { *ii rdist* } CLSTET (*grid*) *x y z*

CLSNODE: returns the ijk-indices of the grid node closest to the spatial location (*x,y,z*) in integer variables *ii*, *ij*, and *ik*. The distance to the node is returned in real variable *rdist*. If an indices window is specified by *ibegin*, *iend*, etc., only nodes within the indices window are searched. Otherwise, the entire grid is searched. If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *ii ij ik rdist* } CLSNODE (*grid*) *x y z ibegin iend jbegin jend kbegin &*
kend

CLSPLOD: returns the number in variable of the pressure-loaded face (defined by **PLOD**) closest to the user-specified location defined by spatial coordinates (*x,y,z*). The distance to the center of the pressure-loaded face is returned in real variable *rdist*. If more than one pressure-loaded face is found at distance *rdist* from (*x,y,z*), then up to three face numbers are returned in integer variables *j1*,*j,2* and *j3*. If the closest pressure-loaded face is unique, then *j2* and *j3* are returned with value 0.

#GET { *j1 j2 j3 rdist* } CLSNODE *x y z*

CLSSHEL: same as CLSNODE except the indices are for the closest SHEL element and the distance is to the element's center. Lyrbeg, lyrend denote the layer range for the shell element. If *lyrnam* is input, the clossest shell using layer definition *lyrnam* will be returned

#GET { *ii rdist lyrbeg lyrend* } CLSSHEL (*grid*) *x y z (lyrnam)*

CRDELEM: same as CRDNODE except the coordinates of an element are returned in real variables *rxcrd, rycrd,* and *rzcrd.* If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *rxcrd rycrd rzcrd* } CRDELEM (grid) *i j k*

CRDGLOB: is the inverse of the CRDLOCL function. The local coordinates are provided by the user and the global coordinates are returned by the function.

#GET { *xg yg zg* } CRDGLOB *axisname rcrd1 rcrd2 rcrd3*

Coordinate Data (continued)

CRDLOCL: returns the coordinates(*rcrd1, rcrd2, rcrd3*) in local system *axisname* for a point defined in global coordinate space (*xg, yg, zg*). The **AXIS** command must have been used to define *axisname* prior to use of this function. If *axisname* is a Cartesian system, the three coordinates returned are the local x, y, and z coordinates, respectively. If *axisname* is cylindrical, they are the radial, circumferential, and axial coordinates. If *axisname* is spherical, they are the radial, theta, and phi coordinates.

#GET { *rcrd1 rcrd2 rcrd3* } CRDLOCL *axisname xg yg zg*

CRDNODE: returns the coordinates of grid node with ijk-indices specified by *i, j,* and *k* in real variables *rxcrd, rycrd,* and *rzcrd.* If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *rxcrd rycrd rzcrd* } CRDNODE (grid) *i j k*

CRDTET: same as CRDNODE except the coordinates of a tet element are returned in real variables *rxcrd, rycrd* and *rzcrd.* If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *rxcrd rycrd rzcrd* } CRDTET (grid) *ii*

DISTELEM: This function is the same as DISTNODE except that integer inputs for either P1 or P2 are interpreted as element i-, j-, and k-indices and the coordinates used for the point are the position of the element center.

#GET { *rdist rdelx rdely rdelz* } DISTELEM *ix1 ix2 jy1 jy2 kz1 kz2*

DISTNODE: returns the total distance and the delta of the x-, y-, and z-coordinates between two spatial locations, P1 and P2. Point P1 is defined by input parameters *ix1, jy1,* and *kz1.* If all three are integers, they are interpreted as the i-, j-, and k-indices of a node. If any are real numbers they are interpreted as the x,y, and z-coordinates of P1. The same approach is used to determine the coordinates of P2, based on input parameters *ix2, jy2,* and *kz2.* The distance between the two locations is returned in real variable *rdist.* The values defined by $(x_2 - x_1)$, (y_2-y_1), and (z_2-z_1) are returned in real variables *rdelx, rdely,* and *rdelz,* respectively.

#GET { *rdist rdelx rdely rdelz* } DISTNODE *ix1 ix2 jy1 jy2 kz1 kz2*

GCONLOC: returns the global ijk indices corresponding to a local general connectivity node number. The GCON USE command should be issued prior to using this command to specify which name of the general connectivity mapping definition to use.

#GET { *iloc jloc kloc* } GCONLOC *gconindex*

INDXLOC: returns the global ijk indices corresponding to a global flex index number. *option* can be set to either NODE or ELEM to find either node or element indices. Default is NODE.

#GET { *iloc jloc kloc* } INDXLOC *index option*

Coordinate Data (continued

LOCINDX: returns the global flex index number corresponding to the global ijk indices. *option* can be set to either NODE or ELEM to find either node or element index. Default is NODE.

#GET { *index* } LOCINDX *iloc jloc kloc option*

MATBOUND: returns the spatial coordinates or IJK-indices that bound the specified material that has been assigned to continuum elements. If the IJK option is specified, the function returns six integer values that identify the IJK-indices bounds. If the XYZ option is specified, the function returns six real values that define the bounding x-, y-, and z-coordinates.

#GET { *ibegin iend jbegin jend kbegin kend* } MATBOUND IJK *matname*

or

#GET { *xbegin xend ybegin yend zbegin zend* } MATBOUND XYZ *matname*

Elemental Data

COUNT: this option returns the total number of objects of type *type* in the model. Valid options for *type* are LINE, BAR, BEAM, MEMB, SHEL, TET, HEX.

#GET { *nobj* } COUNT *type*

CLSELEM: same as CLSNODE except the indices are for the closest continuum element and the distance is to the element's center.

#GET { *ii ij ik rdist* } CLSELEM (*grid*) *x y z ibegin iend jbegin jend kbegin kend*

CRDELEM: same as CRDNODE except the coordinates of an element are returned in real variables *rxcrd, rycrd,* and *rzcrd.* If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *rxcrd rycrd rzcrd* } CRDELEM (grid) *i j k*

CRDTET: same as CRDNODE except the coordinates of a tet element are returned in real variables *rxcrd, rycrd* and *rzcrd.* If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *rxcrd rycrd rzcrd* } CRDTET (grid) ii

DISTELEM: same as DISTNODE except that integer inputs for either P1 or P2 are interpreted as element i-, j-, and k-indices and the coordinates used for the point are the position of the element center.

#GET { *rdist rdelx rdely rdelz* } DISTELEM *ix1 ix2 jy1 jy2 kz1 kz2*

ELEMNUM: this option returns the element number corresponding to label *id*. The option is valid for *type* = BAR, BEAM, MEMB.

#GET { *ielm* } ELEMNUM *type id*

MATELEM: returns the name of the continuum material properties assigned to the element with ijk-indices defined by *i, j,* and *k* in character variable *cmat*. If the optional parameter *erodeopt* is set to ERODE, the word "eroded" is returned for eroded elements. If *erodeopt* is set to ERODEMAT, the name of the material before erosion is returned for an element that has been eroded.

#GET { *cmat* } MATELEM *(erodeopt) i j k*

MODELEM: when the **MODS** command group is used to erode continuum or other element types, this option returns the total number of eroded elements of each specific element category up to this point in the analysis.

#GET { *nelemcont nelemshel nelembm3d nelemmbrn nelembar* } MODELEM

LABEL: this option returns the label *id* corresponding to element number *ielm*. The option is valid for *type* = BAR, BEAM, MEMB.

#GET { *id* } LABEL *type ielm*

Nodal Information

CLSNODE: returns the ijk-indices of the grid node closest to the spatial location (*x,y,z*) in integer variables *ii*, *ij*, and *ik*. The distance to the node is returned in real variable *rdist*. If an indices window is specified by *ibegin*, *iend*, etc., only nodes within the indices window are searched. Otherwise, the entire grid is searched. If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *ii ij ik rdist* } CLSNODE (*grid*) *x y z ibegin iend jbegin jend kbegin & kend*

CRDNODE: returns the coordinates of grid node with ijk-indices specified by *i*, *j*, and *k* in real variables *rxcrd*, *rycrd*, and *rzcrd*. If multiple grids are present such as in Review, use the optional *grid* parameter to name the grid of interest.

#GET { *rxcrd rycrd rzcrd* } CRDNODE (grid) *i j k*

DELPATH: returns the path length defined by the spatial locations of a row of nodes that are contiguous in the i, j, or k direction. The distance along the path is returned in real variable *rdist*. The path is defined by the input parameters *ibegin*, *iend*, etc. To form a valid path, if *ibegin* *iend*, then *jbegin* must equal *jend* and *kbegin* must equal *kend*. Similar constraints hold for the cases of *jbegin* *jend* and *kbegin* *kend*.

#GET { *rdist* } DELPATH *ibegin iend jbegin jend kbegin kend*

DISTNODE: returns the total distance and the delta of the x, y, and z coordinates between two spatial locations, P1 and P2. Point P1 is defined by input parameters *ix1*, *jy1*, and *kz1*. If all three are integers, they are interpreted as the i-, j-, and k-indices of a node. If any are real numbers they are interpreted as the x-, y-, and z-coordinates of P1. The same approach is used to determine the coordinates of P2, based on input parameters *ix2 jy2 kz2*. The distance between the two locations is returned in real variable *rdist*. The values defined by $(x_2 - x_1)$, $(y_2 - y_1)$, and $(z_2 - z_1)$ are returned in real variables *rdelx*, *rdely*, and *rdelz*, respectively.

#GET { *rdist rdelx rdely rdelz* } DISTNODE *ix1 ix2 jy1 jy2 kz1 kz2*

Material Information

MATBOUND: returns the spatial coordinate or IJK-indices that bound the specified material that has been assigned to continuum elements. If the IJK option is specified, the function returns six integer values that identify the IJK-indices bounds. If the XYZ option is specified, the function returns six real values that define the bounding x-, y-, and z-coordinates.

#GET { *ibegin iend jbegin jend kbegin kend* } MATBOUND IJK *matname*

or

#GET { *xbegin xend ybegin yend zbegin zend* } MATBOUND XYZ *matname*

MATELEM: returns the name of the continuum material properties assigned to the element with ijk-indices defined by *i, j,* and *k* in character variable *cmat*. If the optional parameter *erodeopt* is set to ERODE, the word "eroded" is returned for eroded elements. If *erodeopt* is set to ERODEMAT, the name of the material before erosion is returned for an element that has been eroded.

#GET { *cmat* } MATELEM *(erodeopt) i j k*

MATPROP: returns the value of the continuum material property *propname* for material *matname*. Available names are listed on the flxprt file after the MATR input section.

#GET { *pval* } MATPROP *matname propname*

MATNUM: returns the material number for material *matname*. Returns 0 if the material has not been defined

#GET { *mnumbr* } MATNUM *matname*

MATTET: returns the name of the continuum material properties assigned to the tet element with index i in character variable *cmat*. If the optional parameter *erodeopt* is set to ERODE, then the word 'eroded' is returned for eroded elements. If *erodeopt* is set to ERODEMAT, then the name of the material before erosion is returned for an element that has been eroded.

#GET { *cmat* } MATTET *(erodeopt) i*

Graphical Information

MOVIE: when the **FILM** command group in **Review** is used to play back movie animations, this option returns the number of frames in the current animation in the integer variable *inframe*.

#GET { *inframe* } MOVIE

VIEW: returns the values of the eye, view, and vert (vertical) points being used to define the display orientation of 3D models. These values are returned in real variables *rxeye, ryeye,* etc.

#GET { *rxeye ryeye rzeye rxvert ryvert rzvert rxview ryview rzview*} VIEW

WINDOW: returns the bounding spatial coordinates used for setting the displayed image size for the last plot of the model. If the GRPH WNDO subcommand was used, the values returned are the parameters that were specified on the WNDO subcommand. If not, the values returned are the bounding geometric values of the image being plotted. The results are returned in real variables *rxbegin, rxend,* etc.

#GET { *rxbegin rxend rybegin ryend rzbegin rzend* } WINDOW

MODELBOUNDS: returns the bounding spatial coordinates of the physical model. These coordinates do not include extra line draws that may have been added to the plot. The results are returned in real variables *rxbegin, rxend,* etc.

#GET { *rxbegin rxend rybegin ryend rzbegin rzend* } MODELBOUNDS

Mods Information

MODELEM: when the **MODS** command group is used to erode continuum or other element types, this option returns the total number of eroded elements of each specific element category up to this point in the analysis.

#GET { *nelemcont nelemshel nelembm3d nelemmbrn nelembar* } MODELEM

MODVOL: when the **MODS** command group is used to erode continuum elements, this option returns the total eroded volume up to this point in the analysis (in variable *rvoltotal*) and the incremental eroded volume resulting since the last call to the **MODS** primary command (in variable *rvolinc*).

#GET { *rvoltotal rvolinc* } MODVOL

Load Information

AIRBLAST: when the **ALBS CWEP** command in **NLFLex** is used to define a traveling CONWEP pressure function, this option returns several statistics related to the applied air pressures. The basice form returns data in *rtamin* (minimum arrival time), *rtamax* (maximum arrival time), *rpmax* (maximum peak pressure), *rtdur* (time duration of positive phase), *rpneg* (peak negative phase pressure) and *rndur* (time duration of negative phase). If the IMPULSE option is set, the command returns the maximum incident and reflected impulse (positive phase) in *rinc_imp* and *rref_imp*, respectively. *rneg_imp* will contain the maximum impulse for the negative phase.

#GET { *rtamin rtamax rpmax rtdur rpneg rndur* } AIRBLAST

or

#GET { *rinc_imp rref_imp rneg_imp* } AIRBLAST IMPULSE

CLSPLOD: returns the number in variable of the pressure-loaded face (defined by **PLOD**) closest to the user-specified location defined by spatial coordinates (*x,y,z*). The distance to the center of the pressure-loaded face is returned in real variable *rdist*. If more than one pressure-loaded face is found at distance *rdist* from (*x,y,z*), then up to three face numbers are returned in integer variables *j1*,*j2*, and *j3*. If the closest pressure-loaded face is unique, then *j2* and *j3* are returned with value 0.

#GET { *j1 j2 j3 rdist* } CLSNODE *x y z*

PMAPBLAST: when using the **ALBS PMAP** command in **NLFlex** to define applied pressure-time histories from a CFD code, this option returns several statistics related to the applied air pressures. The command returns data in *rtmin* (earliest arrival time), *rtmax* (maximum time of any record), *rpmin* (minimum peak pressure), *rpmax* (maximum peak pressure), *rimin* (minimum peak impulse of all PMAP loads), and *rimax* (maximum peak impulse of all PMAP loads).

#GET { *rtmin rtmax rpmin rpmax rimin rimax* } PMAPBLAST

Character Information

CHARLEN: returns the length (number of characters excluding trailing blanks) of the character string *string* in integer variable *ilength.*

#GET { *ilength* } CHARLEN *string*

CHARLOC: returns the location of *substring* within *string*. The location is returned in integer variable *ilocate.* Returns 0 if *substring* is not present in *string*. If *nbegin* and *nend* are included, these values define the beginning and ending character positions within *string* to search.

#GET { *ilocate* } CHARLOC *string substring nbegin nend*

CHARSET: returns a new character string in variable *cnewstring* created by inserting *setstring* within *string* at the character locations defined by *nbegin* and *nend*.

#GET { *cnewstring* } CHARSET *string setstring nbegin nend*

Miscellaneous Information

CIRCUIT: returns the indices of the starting and ending nodes of a named circuit.

#GET { *istrt iend* } CIRCUIT *circuitname*

COMPWNDO: returns the current size of the computational window as either IJK or XYZ bounds. The computational window is set with the WNDO command and grows with each timestep. *option* is either IJK or XYZ to indicate the returned bound type.

#GET { *il/xmin i/xmax j/ymin j/ymax k/zmin k/zmax* } COMPWNDO *option*

ECHOTIME: returns the difference between model time and actual time of the time history computed using the **ECHO** command in **PZFlex**. **ECHO** computes a time history of the return signal at the top of the model resulting from a distant reflector using a Kirchhoff extrapolation technique. The computed time history can be saved during the analysis. The time associated with the saved record, however, is model time rather than physical time. Time for this record should be time-shifted by the difference between model and physical time for the extrapolated signal, which is returned in real variable *rtime*.

#GET { *rtime* } ECHOTIME

FINDFILE: returns whether or not the disk file *filename* exists. The result is returned in character variable *cfind*. *cfind* is set to "found" if the *filename* exists. If not, *cfind* is set to "notfound."

#GET { *cfind* } FINDFILE *dataname*

MMAPINFO: returns the coordinate bounds and number of materials from an external tetrahedral mesh file. (only available for NASTRAN files) *filename* is the name of the external file. If *rootname* is specified 6 new symbol variables for each material in the external file are created to store the minimum and maximum bounds of each material. The new variables are named *rootname_xmnN, rootname_xmxN, rootname_ymnN,* etc., where *N* is a material number. If *axisname* is specified, the bounds are transformed using the previously defined axis system.

#GET { *xmin xmax ymin ymax zmin zmax nmat* } MMAPINFO *filename (rootname)*

NCYCLES: returns the maximum number of subcycling levels for all computational zones in integer variable incyc. (Available onlly after the **PRCS** step in PZFlex, EMFlex, etc.)

#GET { *incyc* } NCYCLES

Miscellaneous Information (continued)

RANDOM: returns a random number based on the integer parameter *iseed* in real variable *rvalue*. The optional *iseed* parameter should be specified only on the first occurrence of the RANDOM function for which the specific *iseed* value is to be used. Succeeding calls to RANDOM make use of the same *iseed* value. If *iseed* is never specified, *iseed* =1 is assumed.

#GET { *rvalue* } RANDOM (*iseed*)

TABLMATR: Returns number of materials on the site input file. Also creates symbol variables rootname1, rootname2,... with the material names. This option is implemented for the file format with the NAME record, but not for the format without NAME.

#GET {*nmat* } TABLMATR *filename rootname (fileformat)*

TABLSIZ: Returns coordinates and numer of cells on a site input file. *fileformat* is asci or bin default = asci. These are in the coordinate system of the file, not shifted or transformed into model coordinates.

#GET { *xmin xmax ymin ymax zmin zmax nx ny nz* } TABLSIZ *filename (fileformat)*

TERM: returns the current state of the terminal input and output contol variables (i.e., those last defined by a **TERM** primary command) in character variables *cinput* and *coutput*, respectively. These variables contain the value of ON or OFF.

#GET { *cinput coutput* } TERM

Symbol Information

SYMBTYPE: returns the symbol type: real, int, char or text of *symbolname*

#GET { *itype* } SYMBTYPE *symbolname*

ROOTMAX: returns the largest integer suffix of a symbol beginning with *rootname*. E.g. if X1, X2, X3 are definded, *maxindx* would be 3.

#GET { *maxindx* } ROOTMAX *rootname*

Review Information

In **Review**, information about snapshot and time history or other x-y data files which have been read in and assigned to a local file name can be accessed with the following functions:

CURVCORD: returns the x-, y- and z-coordinates assigned to record *irecord* on *localfilename.* The xyz-coordinates are returned in real variables *xc*, *yc* and *zc* , respectively.

#GET { *xc yc zc* } CURVCORD *localfilename irecord*

CURVCROS: similar to CURVMINS except it returns the curve locations that cross the y=*value* line. Setting *value*=0.0 would return the zero crossings for a curve.
Note that the curve sampling points are interpolated to compute the actual crossing locations.

#GET { *incros* } CURVCROS *localfilename icurve value rootname nmax begin end*

CURVDELX: returns the difference x(2)- x(1) for *localfilename* in real variable *rdelx.* For uniformly sampled data, this is the sampling increment. For time history data, this is the time increment between the first two time points. All time histories in a file have the same time sampling.

#GET { *rdelx*} CURVDELX *localfilename*

CURVIJKL: returns the ijk-indices and label for record *irecord* on *localfilename.* The ijk-indices are returned in integer variables *ii*, *ij,* and *ik* , respectively. The curve label is returned in character variable *clabel.*

#GET { *ii ij ik clabel* } CURVIJKL *localfilename irecord*

CURVMAX: same as CURVMIN except it returns the maximum value.

#GET { *rxvalue ryvalue ipoint* } CURVMAX *localfilename icurve begin end*

CURVMAXS: same as CURVMINS except it returns the local maximums.

#GET { *inmaxs* } CURVMAXS *localfilename icurve rootname nmax begin end*

CURVMIN: returns the x and y values of the point on curve number *icurve* on *localfilename* whose y value is a minimum for the range *begin* to *end*. The values are returned in real variables *rxvalue* and *ryvalue*. The time sampling point is returned in integer variable *ipoint*..

#GET { *rxvalue ryvalue ipoint* } CURVMIN *localfilename icurve begin end*

Review Information (continued)

CURVMINS: returns the x- and y- values of the local minimums for curve number *icurve* on *localfilename* for the range *begin* to *end*. The number of local minimums found is returned in integer variable *inmins* up to a maximum of *nmax* points (default: *nmax*=10). The actual x- and y-values defining the minimum are assigned to variables with the names *rootname*x1, *rootname*y1, *rootname*x2, *rootname*y2, etc.

#GET { *inmins* } CURVMINS *localfilename icurve rootname nmax begin end*

CURVNVAL: returns the number of x-y (time, data) values for each curve for the file *localfilename* in integer variable *inval*.

#GET { *inval*} CURVNVAL *localfilename*

CURVNREC: returns the number of records (curves) on *localfilename* in integer variable *inrec*.

#GET { *inrec*} CURVNREC *localfilename*

CURVRECD: returns the record number of a record on *localfilename* matching the provided identifier information in integer variable *irecd*. The identifier information is provided to specify the ijk indices and curve label of the record of interest. *identifier* may be any of: LABL, I, J, or K. *idvalue* is the value for *identifier*. For example, the identifier information: LABL XVEL I 21 J 16 K 2 requests that *irecd* be set to the time history record number containing the x-velocity for the node with ijk of (21, 16, 2). Any combination of identifiers may be specified. The first record that matches the specified indentifier information is selected. *irecd* = 0 indicates that no matching record was found.

#GET { *irecd* } CURVRECD *localfilename identifier idvalue* (repeat as needed)

CURVTITL: returns the title for file *localfilename* in text variable *ctitle*.

#GET { *ctitle* } CURVTITL *localfilename*

CURVTSHIFT: returns timeshift information for a specific time history number *icurve* on *localfilename* and also provides the minimum and maximum time-shift values for all curves within *localfilename*. The time-shift values are returned in real variables *rshift, rshifmin,* and *rshiftmax* . Time-shift values are usually non-zero only for time histories created using extrapoltion functions.

#GET { *rshift rshiftmin rshiftmax* } CURVTSHIFT *localfilename icurve*

CURVTYPE: returns the type of file *localfilename* in text variable *ctype*.

#GET { *ctype* } CURVTYPE *localfilename*

Review Information (continued)

CURVVALU: returns the x and y values for a requested x (abscissa) value for curve number *icurve* on *localfilename*. The values are returned in real variables *xvalue* and *yvalue*. Each curve is actually discretely sampled and is represented by a finite number of x(i)-y(i) data pairs. The *option* parameter allows the user to control how *xvalue* and *yvalue* are determined. *option* can be set to: CLOSE, LOW, HIGH, and INTERP. If *xrequest* falls between x(n) and x(n+1), then if option is set to CLOSE, *xvalue* and *yvalue* are returned as x(n),y(n) or x(n+1),y(n+1), depending on whether x(n) or x(n+1) is closer to *xrequest*. If *option* is set to LOW, then *xvalue* and *yvalue* are returned as x(n),y(n). If option is set to HIGH, then *xvalue* and *yvalue* are returned as x(n+1),y(n+1). If option is set to INTERP, then *xvalue* is equal to *xrequest* and *yvalue* is interpolated from the discrete sampling values. The specific sampling point whose values are returned as *xvalue,yvalue* is returned in integer variable *ipoint*. When *option* is set to INTERP, *ipoint* is returned as the lowest numbered sampling value used to evaluate *xvalue,yvalue*.

#GET { *xvalue yvalue ipoint* } CURVVALU *localfilename icurve xrequest option*

Note: The next seven options return minimum or maximum value information for a curve contained on a local file; the values *begin* and *end* define the range of a curve to search. If they are not input, the min or max values for the entire curve are returned. If they are input as integer quantities (i.e., with no decimal point), they are assumed to be the beginning and ending sampling points from the curve over which to evaluate the min or max. If they are input as real values (i.e., with a decimal point), they are assumed to be the values of the x range of the data over which the min or max is desired. For time histories, this would be the beginning and ending time, and for frequency data this would be the beginning and ending frequency, etc. See examples below.

CURVWNDO: returns the x width and left and right x-values of the portion of curve number *icurve* on *localfilename* whose y value exceeds *tvalue*, a user-defined threshold limit. The form of *tvalue* is defined by *value_option*, which currently must be set to DB, i.e., *tvalue* is specified in terms of the number of db down from the peak y value for the portion of the record being searched. The *begin* and *end* parameters are similar to the preceding four functions. If not input, the entire record is searched. If input, the search is limited to the x region defined by *begin* and *end*. If *search_option* is set to OUT, the search for the points at which the curve's y value drops below *tvlaue* is from the peak point in the search range out to its edges. If *search_option* is set to IN, the search is performed from the edges in toward the peak point. The left and right x values of the portion of the curve that exceeds *tvalue* are returned in real variables *rleftx* and *rightx*. The x width (*rightx-rleftx*) is returned in real variable *rwidthx*. The character variable *cerrstatus* returns the error indicator for this function. It is set to: NONE, LEFT, RIGHT or BOTH. If NONE, the search for the left and right cutoff points was successful. If LEFT, no left cutoff point was found before the left side of the search range was encountered. If RIGHT, no right cutoff point was found before the right side of the search range was encountered. If BOTH, neither a left nor right side point was found to fall below the *tvalue* specified.

#GET { *rwidthx rleftx rightx cerrstatus* } CURVWNDO *localfilename icurve &*
tvalue value_option search_option begin end

Review Information (continued)

CURVXMIN: returns the minimum x value of curve number *icurve* on *localfilename* over the range *begin* to *end*. The value is returned in real variables *rxvalue* .

#GET { *rxvalue* } CURVXMIN *localfilename icurve begin end*

CURVXMAX: same as CURVXMIN except it returns the maximum value.

#GET { *rxvalue* } CURVXMAX *localfilename icurve begin end*

SNAPMAX: returns the total number of snapshots times that have been read in for *localfilename.* The number of snapshot times is returned in integer variable *nsnap*.

#GET { *nsnap* } SNAPMAX *localfilename*

The #SET Option

The #SET option provides the reverse functionality of the #GET option. It sets the value of a data manager array location equal to the value of the specified symbol variable. It has the following form:

SYMB #SET value dataname i j k

where:

value = the value to be assigned to the data manager array location.
dataname = the name of the data manager array
i,j,k = the i-, j-, and k-indices of the location within the data array whose value will be set with this command.

For example, the following command assigns the value of data manager array DAT1(3,9,4) to be 3.5:

SYMB #SET 3.5 DAT1 3 9 4

Note that this command may be used to set the value of any integer, real, or character data array.

The #ADD Option

The #ADD option provides similar functionality to the #SET option. It modifies the value of a data manager array location by adding the specified value. It has the following form:

SYMB #ADD value dataname i j k

where:

value = the value to be added to the data manager array location.
dataname = the name of the data manager array
i,j,k = the i-, j- and k-indices of the location within the data array whose value will be changed with this command.

For example, the following command will add 3.5 to the value of data manager array DAT1(3,9,4);

SYMB #ADD 3.5 DAT1 3 9 4

Note that this command may be used to set the value of any integer or real data array.

The #MSG Option

The #MSG option allows messages to be sent to the Interactive User (IU) at the terminal screen or written to a disk file. It also allows the IU to assign a value to a Symbol variable through the keyboard. Input from, and output to, the IU is distinct from the standard input and output units of the code. This allows an interactive job to be configured in which the IU sees only messages specifically routed to the terminal. At certain points in the processing of a group of input commands, the IU can be queried to provide data or be informed of the status of the job. The bulk of the output from the code need not be sent to the IU in order to allow interaction between the code and the user. The #MSG option can take either of two forms depending on whether information flow is one way or two ways:

Form 1: Messages are written and no response from the Interactive User occurs.

SYMB #MSG *nl*
(*nl* lines of messages must follow the **SYMB** statement. They are displayed on the terminal)

SYMB #MSG *nl* > *filename*
(*nl* lines of messages must follow the **SYMB** statement. They are written to *filename*)

SYMB #MSG *nl* >> *filename*
(*nl* lines of messages must follow the **SYMB** statement. They are appended to *filename*)

SYMB #MSG *nl* &> *filename*
(*nl* lines of messages must follow the **SYMB** statement. They are written to *filename* AND displayed on the terminal)

SYMB #MSG nl &>> filename
(*nl* lines of messages must follow the **SYMB** statement. They are appended to *filename* AND displayed on the terminal)

Form 2: Messages are written to the terminal screen which query the Interactive User to provide keyboard input to define the value of a Symbol variable.

SYMB #MSG *nl vname vtype* <constraints>
(*nl* lines of messages must follow the **SYMB** statement. They are displayed on the terminal)

where:

nl =the number of lines of messages that follow the **SYMB** statement

vname =the variable name whose value will be assigned by the IU
vtype =the data type of *vname*, any of: REAL, INTEGER, or ALPHA.

<constraints> =constraints upon the value which the IU can assign to *vname*

The variable, *vname*, must correspond to the data type shown.

The message lines which follow the **SYMB** #MSG statement are echoed to the terminal of the IU or written to a disk file. If a Symbol variable name is found in a message line, the value of the variable will be substituted

in place of its name, as illustrated by the following example. When the following inputs statements are processed by the code:

```
symb radius = .000001
symb #msg 1
        THE FEATURE RADIUS = $radius
```

the IU's terminal will display:

```
THE FEATURE RADIUS = .000001
```

The constraints upon the value assigned to the variable can be of two forms. The first, termed list constraints, applies to ALPHA type variables (those which contain character data). These constraints start with the key word LIST, followed by a list of allowable responses. Several examples are:

```
symb #msg 1 answer alpha list yes no
DO YOU WANT TO CONTINUE WITH THE ANALYSIS?

symb #msg 1 mattype alpha list slcn phot oxid
WHAT FEATURE MATERIAL IS TO BE USED IN THE ANALYSIS?
```

The first example will ask the IU to reenter his response until either YES or NO is entered. In the second example, the only acceptable responses by the IU are SLCN, PHOT or OXID. The user can abbreviate a list item by typing only the characters needed to uniquely identify a list item. For example, Y could be typed in for YES, N for NO, etc. The second type of constraint, is termed a conditional constraint. These constraints can be applied to numeric data. They take the form of: *op1 value1 op2 value2* etc., where *op* is one of the conditional operators EQ, NE, GT, GE, LT or LE. Their use is shown in the following examples:

```
symb #msg 1 epsilon real gt 0.
WHAT IS THE EPSILON OF THE MATERIAL?
```

In this example, if the IU provides a value for EPSILON which is less than or equal to 0.0, he will be requested to reenter a proper value. If the variable being assigned by the **SYMB** #MSG statement has previously been defined, the IU will be informed that the current value is the default value of the variable, and typing a carriage return will leave the variable unchanged.

The #LIB Option

Purpose: this command is a more intelligent form of the SYMB #READ command. It chooses a 'library' file to read based on a date timestamp and the path info provided in a library definition file. It automatically constructs the path name to access the proper file for the user. The path name would default to the 'latest and greatest' version of a file but a user could explicitly specify any date and the code would access the file that was operational on that date. This functionwill support material libraries but can also be used for libraries of any types of functions that are useful in flex/review/build.

Note that the path name to a library can be set in a user's Flex.defaults file but may be manually overridden with an input command as shown below.

Syntax:

SYMB #LIB *libtype* PATH *libpath*
SYMB #LIB *libtype libname date(mmddyy)*

Examples:

symb #lib matr path /d/flex/resources/matr

```
matr
  symb #lib  matr    a570_type1            /* gets the latest version for name= a570_type1
  symb #lib  matr    a360_type2   082207   /* gets the version that was operative on 22aug07
  end
```

Functionality: The code will locate the file: *libtype*.LIBRARY in *libpath. Libpath* should be set to *local* if the library files are in the same directory as the *libtype*.LIBRARY file.

This file contains a list of all the file sets, the date that they became 'the official version', and any path information required to fully construct the pathname and access the data.

matr.library file example:

```
a570_type1.022108      steel/type1
a570_type1.081507      steel/type1
a570_type1.110306      steel/type1
a360_type2.081507      steel/type2
a360_type2.110306      steel/type2
etc.
```

The code will read the matr.lib file, identify the appropriate date stamped file to use, construct the pathname for this file and internally generate a 'symb #read' command for that file.

The #TITLE Option

The #TITLE option allows the IU to interactively provide the job title and id. When this statement is encountered, the IU will be shown what the current title and job id is (if any), and asked to provide title information for the current job.

The #SAVE Option

The #SAVE option allows the current list of symbols and their values to be saved on a specified disk file. They are saved as a group of **SYMB** input statements. If the #TITLE option has been used previously, the file containing the symbol assignments will be preceded by a **TITL** command containing the job title and id. The #SAVE option has two forms:

SYMB #SAVE *fname option echopt*
SYMB #SAVE *

where:

fname = the disk file name, up to 80 characters long
* = the free format default indicator. In this context, the user is queried to provide *fname*

In the first example, the disk file, *fname*, will be opened, the current symbols and their values will be written to the file, and then the file will be closed. In the second example, the IU is requested to provide the file name. The code will try to open this file. If there is a problem in opening the file, the IU will be told, and then asked to reenter the proper file name. Set *echopt* to noecho to reduce printing

The #READ Option

The #READ option reads in a file containing any program commands or Symbol statements. It opens the file, reads the input data as if the file was a standard input file, and once the eof is found, closes the file and switches back to the previous input file. This option provides a simple way to read portions of the input command stream from various files. Since the file created with the #SAVE option is composed of standard input commands, the #READ option provides a way of reading a file created with the #SAVE option. This allows initializing the values of a group of variable names from a separate data file. The statement has the same two forms as the #SAVE option:

SYMB #READ *fname*

SYMB #READ *

The #LIST Option

The #LIST option allows the user to list all or some subset of the current symbols and there values to the output file. The form of the option is:

SYMB #LIST *pattern*

where *pattern* specifies a common text pattern in the name of all the variables which are to be listed. If *pattern* is not input, all symbols are listed.

The wildcard character, @, is allowed in the *pattern* variable. Several examples are: if *pattern* = I@, all symbols which begin with I will be listed; for *pattern* = K@GRID, any symbols beginning with K and ending with GRID will be listed; and if *pattern* = @, all symbols will be listed.

The #MODIFY Option

The #MODIFY option allows for the scaling or shifting of the numeric values assigned to a group of symbol variables with one statement. The form of the option is:

SYMB #MODIFY *option pattern value echoopt*

where *option* is either SCAL or SHFT, *pattern* specifies the common text pattern in the name of all the variables which are to be modified, and *value* is the value used to scale or shift the specified variables.

The wildcard character, @, is allowed in the *pattern* variable. Several examples are: if *pattern* = X@, all variables that begin with X will be modified; for *pattern* = I@GRID, any variables beginning with I and ending with GRID will be modified; and if *pattern* = @, all variables will be modified. Any character variables are ignored whether or not they match *pattern*. All numeric variables will be shifted or scaled in a fashion consistent with the variables type, i.e., either real or integer. Set *echoopt* to noecho to reduce printing

The #REMOVE Option

The #REMOVE option removes a variable from the current list of symbols. The form of the option is:

SYMB #REMOVE *vname*

where *vname* is the name of the variable to be removed.

The #ECHO Option

The #ECHO option allows the user to suppress most of the printed output which is produced by **SYMB** statements. The form of the option is:

SYMB #ECHO *option*

where *option* is OFF, ON, or VALU. The ECHO option is always ON when the program initiates. If *option* is ON, each **SYMB** statement encountered in the input stream produces 4 lines of output. These include an echo of the **SYMB** statement, the previous value of the variable and the new value of the variable. If *option* is VALU, each **SYMB** statement produces a single line of output noting the new value of the variable. If *option* is OFF, no output is generated by **SYMB** statements.

The #IGNORE Option

The #PRINT option request that the code expand Symbol variables in any input line printed to the Job Output File. It also allows for suppressing print related to Symbol directives. The form of the option is:

SYMB #IGNORE *option*

An option to avoid bad numbers. For option = divide, division by zero produces the numerator. For option = log0, the log of zero is set to zero. The default is #RAISE (see below)

The #RAISE Option

SYMB #RAISE *option*

For option = divide, an error message is produced for division by zero. For option = log0, an error message is produced for log of zero.

The #PRINT Option

The #PRINT option request that the code expand Symbol variables in any input line printed to the Job Output File. It also allows for suppressing print related to Symbol directives. The form of the option is:

SYMB #PRINT *option*

where *option* is OFF, ON, or NONE. The #PRINT option is always OFF when the program initiates. If *option* is set to ON, each **SYMB** statement which contains a Symbol variable will be printed to the Job Output File twice, initially as input, the second time in expanded form. This option is an excellent tool for debugging Symbol scripts. It is typically used in conjunction with the #ECHO ON option. If *option* is set to NONE all print from any Symbol language directive will be suppressed until *option* is redefined by a later #PRINT directive.

The #KILL Option

The #KILL option causes the program to check for the presence of the kill file (Command Override File) for the job at the time it is encountered. If the kill file is found, the current Job Input file is closed, the kill file is opened and the program processes the commands encountered. This option allows the user to choose the point in a Symbol script which provides a proper point for job termination should the need arise to bring the job down before it finishes. The form of the option is:

SYMB #KILL

The #EXIT Option

The #EXIT option causes the program to exit gracefully at the time the command is encountered. The form of the option is:

SYMB #EXIT

The #DELFILE Option

The #DELFILE option deletes a disk file. The form of the option is:

SYMB #DELFILE *filename*

where *filename* is the name of the disk file to delete.

The #SLEEP Option

The #SLEEP option puts the job to sleep for a specified number of seconds. The form of the option is:

SYMB #SLEEP *sleepseconds*

where *sleepseconds* is the number of seconds the process will sleep.

The #SUBMIT Option

The #SUBMIT option submits a process to the system. The process can be run with a WAIT or NOWAIT option. If *option* = WAIT, the current process will wait while the new process is being run. Once it completes, the current process will continue. If *option* = NOWAIT, once the new process is spawned, the current process will continue without pausing. These processes will then be running in parallel. The form of the option is:

SYMB #SUBMIT (option) process_description

where *option* is an optional input parameter. If not input, *option* = WAIT will be assumed for this submittal. *process_description* is the complete command line that represents the new process. For example, on Linux systems, the command:

SYMB #SUBMIT ls flxinp.*

will cause all files in the current directory whose names begin with flxinp to be listed.

The #TEMP Option

The #TMPL option accesses a standard template stored in the tools subdirectory of the FLEXHOME directory (which contains the code validation key for the system). A standard template is one that performs some task based on a set of Symbol variables which must have been set prior to invoking the template. The form of the option is:

SYMB #TMPL *templatename*

which will invoke the template file named *templatename*.temp stored in the tools subdirectory. Certain standard templates are provided with the code. Users may add their own templates also.

The #TOOL Option

The #TOOL option accesses a standard tool template stored in the tools subdirectory of the FLEXHOME directory (which contains the code validation key for the system). A tool template is one that interactively queries a user for the information it needs to perform its function and then performs the desired action, often by invoking a standard template described above. The form of the option is:

SYMB #TOOL *toolname*

which will invoke the file named *toolname*.tool in the tools subdirectory. Certain tool templates are provided with the code. Users may add their own tool templates also.

The #KEY Options

The #KEY options provide an expedient way to quickly define keypoint symbols for standard partition models. The form of the options are:

SYMB #KEYCORD direction indxbegin valuebegin dvalue1 dvalue2 dvalue3 ....

Direction is x,y, or z. *indxbegin* is the first index to be defined. *valuebegin* is the coordinate value at the first index. *dvalue1, dvalue2,...* are coordinate increments to succeeding keypoints. E.g.
SYMB #KEYCORD X 3 10. 5. 5. Would produce $x3=10; $x4=15; $x5=20. Default for *indxbegin* is 1 greater than the last defined.

SYMB #KEYCOPY direction indxbegin indxend ncopy invertoption

Direction is x,y, or z. *indxbegin, indexend* are the symbols to be copied. *Ncopy* is the number of replications and *invertoption* = INVERT to invert coordinate spacing between each succeeding copy. E.g. If $X1=0, $X2= 1, $X3 = 5, then SYMB #KEYCOPY X 1 3 1 invert Would produce $X4 = 9, $X5 = 10.

SYMB #KEYINDX direction indxstart indxstop ivalindxstart spacing minval

Direction is i,j, or k. *indxstart, indxstop* are the symbols to be defined. *Ivalindxstart* is the node number at *indxstart. Spacing* is the coordinate spacing between keypoints and *minval* is the minimum increment between keypoints. Default value for *indxstart is 1.* Default for *indxstop* is the max defined coordinate keypoint index. E.g. symb #keyindx I 1 3 1 2. 2 Will define $I1 = 1; $i2 = $i1 + max(2, (nint ($x2 - $x1)/ spacing)); $i3 = $i2 + max(2,(nint($x3-$x2)/spacing)). The coordinate keypoint symbols, $x1, $x2, $X3 must have been previously defined.

THE **IF**, **ELSEIF**, **ELSE** and **ENDIF** STATEMENTS

Symbol supports the **IF**, **ELSEIF**, **ELSE** and **ENDIF** statements in order to provide structured, conditional branching constructs. An example of the use of these statement is:

```
if ( $n eq 1 ) then
        symb comment = first
elseif ( $n eq $nlast ) then
        symb comment = last
else
        symb comment = middle
endif
```

An IF conditional must begin with an **IF** statement and terminate with an **ENDIF** statement. The basic form of the **IF** statement is:

IF (*datum1 op datum2*) THEN

where *op* is any of the conditional operators: EQ, NE, LT, LE, GT and GE whose meanings correspond to their usage in Fortran. *datum1* and *datum2* can be either numeric or character data. Multiple conditionals may be used in a single **IF** statement by using the AND or OR Boolian operators. The order of conditionals may be controlled through the use of parenthesis. Several examples are:

```
if ( $n eq 1 and $x ne 0.0 ) then
```

and

```
if ( $n eq 1 and ( $x ne 0.0 or $y ne 0.0 ) ) then
```

The form of the **ELSEIF** statement is similar to the **IF** statement.

The **ELSE** and **ENDIF** statements have no other input parameters on the command line.

Once an **IF** statement is encountered, the following input lines down to the terminating **ENDIF** statement are read in and stored as an **IF** procedure which is then invoked. If an **ENDIF** statement is accidentally omitted from the input file, the code will read to the end of the file and terminate with an error.

Nested sets of **IF** statements are allowed.

An example is:

```
if [ $n eq 1 ] then
    if [ $iset eq 1 ] then
            symb comment = first1
    elseif [ $iset eq 2 ] then
            symb comment = first2
    endif
elseif [ $n eq $nlast ] then
    if [ $iset eq 1 ] then
            symb comment = last1
    elseif [ $iset eq 2 ] then
            symb comment = last2
    endif
endif
```

WARNING: 1) Including the **STOP** command within an **IF/ELSEIF** logic block will result in termination of the job at the time the **STOP** command is read. Use **SYMB #EXIT** to provide a code termination statement within an **IF/ELSEIF** logic block.
2) In-line functions on and IF/ELSEIF line are not interpreted. E.g., do not use use commands such as: IF (ABS ($N) EQ 1) THEN

THE **END$** STATEMENT

The **END$** statement identifies a line in the input command file and has the form:

END$ linename

where:

linename =from 1 to 8 character name identifying this line

The program does not require that *linename* be unique. That is, more than one **END$** statement can have the same name.

The **END$** statement is used to identify the location in the input stream to branch to for **GOTO** statements and the ending points of do loops and procedures defined with the **DO** and **PROC** statements, respectively.

THE **PROC** (PROCEDURE) STATEMENT

The **PROC** statement has several options which allow a procedure to be saved, executed or erased. Procedures are a user defined group of commands which a user may want to repeat as a group. Up to 25 procedures may be saved at any one time. Procedures can contain any combination of valid code input lines. Procedures can execute other procedures (i.e., procedures may be nested) up to 25 levels. Procedures may not be executed recursively.

The SAVE Option

A procedure is defined by use of the **PROC** statement with the SAVE option. This option saves the following command lines, up to the next **END$** PROC statement, internally within the program for future use. The form of this statement is:

PROC *procname* SAVE
< procedure command lines >
END$ PROC

where:

procname =a 4 character name uniquely identifying this procedure
(No default)

The ERAS (Erase) Option

A procedure is erased by use of the **PROC** statement with the ERAS option. Its form is:

PROC *procname* ERAS

were:

rocname =the name of the procedure to erase

Executing a Procedure

A procedure may be executed in one of two ways. The user can manually request that the procedure be executed one or more times at a specified point in the input stream or the procedure may be linked to execute at specified times during a **PZFlex** execution by using the RATE option.

A procedure is manually activated by inputing the **PROC** statement with the requested number of times the procedure should be executed specified after the procedure name. The form of the command is:

PROC procname nrepeat

where *procname* = a name of the procedure to execute. No default

nrepeat = an integer value defining the number of times the procedure is to be executed. If *nrepeat* is not input or zero, it defaults to 1.

Linking the execution of a procedure to the time of execution in a PZFlex analysis is a simple way to ensure a desired procedure is activated at a constant time increment throughout an analysis. The RATE may be expressed as either a specified number of timesteps or as a specified time increment. The form of the command is:

PROC procname RATE (option) increment reference_time

where *procname* = a name of the procedure to execute. No default

option = set to either STEP or TIME (if option is omitted, STEP is assumed)

increment = this parameter defines the time increment between the times at which the procedure will be activated.
If option=STEP, then *increment* is input as an integer representing the number of time steps between activations.
If option=TIME, then increment is input as a real number representing the time increment in model units between procedure activations. In increment = 0.0 or default, then the procedure will not be activated during the PZFlex analsysis.

reference_time = This parameter defines the reference time from which the increment will be applied.
If option=STEP, then *reference_time* is the starting time step to which *increment* is applied.
If option=TIME, then *reference_time* is the starting time to which *increment* is applied.
Default = 0.

Examples

Some examples of **PROC** statement usage are:

```
proc        run save   /* save the following 9 input lines as a procedure named "run"
exec        500
data
        out dat1
        out dat2
        end
grph
        plot dat1
        end
end$        proc                    /* this statement identifies the end of the procedure


proc        run                     /* execute the procedure 1 time

proc        run 20                  /* execute the procedure 20 times

proc        run eras                /* erase the procedure
```

```
proc        graph save          /* save the following 3 input lines as a procedure named GRAPH

grph
        plot dat1
        end
end$        proc                    /* this statement identifies the end of the procedure

proc        grph rate 1000          /* execute the procedure every 1000 timesteps

proc        graph rate time 2.5e-9  /* execute the procedure every 2.5e-8 seconds
```

THE **DO** STATEMENT

The equivalent of Fortran 77 do loops are provided through the use of the **DO** statement. It is used in conjunction with an **END$** statement to define the beginning and end of a do loop. Do loops may be nested up to 10 levels. The form of the **DO** statement is:

DO doname dovariable nbegin nend ninc printoption
    < do loop command lines >
END$ doname

where:

doname = is the line name of the **END$** statement which denotes the end of this do loop. No default.

dovariable = is the name of an integer symbol variable which will be defined and incremented during the execution of the do loop. No default

nbegin = is the starting value of *dovariable.* No default

nend = is the ending value of *dovariable.* No default

ninc = is the value by which *dovariable* is incremented during each loop (Default = 1)

printoption = 4 character parameter which allows the user to suppress much of the output generated during a do loop. May be any of: ON, OFF
Default is ON for the outermost **DO** statement. For nested loops, the default is the value of *printoption* for the next outermost loop.

If set to ON, standard output is generated.

If set to OFF, echoing of all input commands is suppressed as well as all output generated by **SYMB**, **GOTO**, **PROC**, **DO** , **IF**, **ELSEIF** , **ELSE**, **ENDIF** and **END$** statements.

Some examples of **DO** statement usage are shown here. The first example is a single 3 line do loop which assigns 100 variable names with the integer values 1 to 100.

```
symb    n = 0              /* initialize n to 0
do      loop1 I 1 100      /* define and execute a 3 line do loop
symb    n$I = $I
end$    loop1              /* the end statement of this do loop
symb    #list              /* list symbols after the do loop: n1=1, ... , n100=100, I=101
```

The second example shows a group of 5 nested do loops. Note that nested loops may end with the same **END$** statement.

```
symb    n = 0              /*initialize n to 0

do      loop1 I 1 5
do      loop1 J 1 5
do      loop1 K 1 5
do      loop2 L 1 5
do      loop2 M 1 5
symb    n = $n + 1
end$    loop2
end$    loop1
symb    #list      /* list all symbol values after loop  n=3125, I=6, J=6, K=6, L=6, M=6
```

A large number of passes though a do loop can generate a significant amount of output. This is because, by default, each input line is echoed to the output file, the previous and present value of each symbol variable assignment is output, etc. As shown, the above example will produce over 19,100 lines of output. The user has control over the amount of output generated by **SYMB** statements and by do loops. For example, as discussed previously for the **SYMB** statement, preceding the above example with the statement:

SYMB #ECHO VALU

reduces the number of lines of output produced to about 9,700 since for each execution of the **SYMB** statement embedded within the do loop, only a single line of output will be produced showing the present value of the variable N. Alternatively, the statement:

SYMB #ECHO OFF

will reduce the number of lines of output to about 6600 since no output will be generated from the embedded **SYMB** statement. To suppress all output generated by this do loop, the *printoption* parameter on the **DO** statement can be set to OFF. For example, if the first **DO** statement in the above example were replaced by the following:

DO LOOP1 I 1 5 OFF

only 46 lines of output would be generated by the example.

THE GOTO STATEMENT

The ability to branch to any other location in the input command file is provided by the **GOTO** statement. The **GOTO** statement names a line name to branch to in the current input stream. This line name is defined with a matching **END$** statement. The **GOTO** statement can have two forms:

The unconditional branch:

GOTO linename

and the conditional branch:

GOTO linename IF datum1 op datum2

where:

linename =	the line name to branch to
datum1, *datum2* =	two data values (either numeric or character data)
op =	a conditional operator (any of EQ, NE, LT, LE, GT and GE)

The **GOTO** statement will branch to the first **END$** statement following the **GOTO** statement with a line name which matches the **GOTO** line name. If the current input file eof (end-of-file) is found before the matching **END$** statement is located, searching continues at the start of the input file. If a matching branch point is not found after a complete pass through the input file. The job terminates with an error. If a matching **END$** statement is found, the branch is complete and input commands following the **END$** statement are processed in the normal fashion.

If a **GOTO** statement is placed within a procedure the matching **END$** statement must occur within the procedure, i.e., the procedure is considered to be the current input file. If not found, the job terminates with an error. If a **GOTO** statement is encountered within the range of a **DO** loop, the matching **END$** statement is searched for following the **GOTO** but within the range of the complete do loop. If the end of the do loop is found, searching begins again at the beginning of the loop. If no matching **END$** statement is encountered within the complete range of the loop, it is assumed that the user is branching out of the do loop and that the do loop is complete. Searching begins for the matching **END$** statement following the **DO** loop in the procedure or input file which contains the loop.

2 COMMAND DEFINITIONS

DESCRIPTION OF COMMAND DEFINITIONS

The command definitions use bold capital letters for all primary commands and subcommands. Input arguments are shown in lower case italic letters. Data types of input arguments are indicated as: (F) = floating point real number, (I) = integer number, (C) = character string and (X) = a generalized construct which may take different forms and be composed of several arguments and data types. Actual character string inputs are shown with all capital letters.

NOTE:	**All character string inputs to the program, including all commands and input parameters, should be lower case. Only Symbol variable names and values may be upper case.**

A compact list of all commands and arguments is presented first, followed by a detailed description of each command defining its purpose, whether it is required for a typical analysis, if it can be used more than once in the input stream, and whether or not there are any order constraints on its use. Primary commands are presented in alphabetical order. Subcommands follow each primary command in the order listed for the command group. The compact list and the primary command pages indicate which codes the command is applicable to. If no code is listed, the command group applies to all programs.

Input parameters shown in parentheses are optional and they need not be placed on the command line. If an optional parameter is not input, an * (default indicator) is not needed to mark its position on the command line.

COMMENT LINES

Comment lines may be placed at any point in the input data stream where a primary or subcommand is expected. A comment line is identified by placing a "**C**" as the first input parameter on the line followed by at least one blank space. The program will read the comment line, write it to the Job Output File, and then continue processing the job. Also, any command input line may have a comment field appended to it by preceding the comment field with a /*.

COMMAND STRUCTURE

PZFlex uses the free-format input form described below to define the problem of interest. A system of keyword commands are used to direct the flow of job processing. All keywords are strings of from 1 to 4 characters with no abbreviations allowed. There are two levels of keywords used. The first level are called primary commands (PCOM), while those in the second level are called subcommands (SCOM). A PCOM may have any number of subcommands associated with it, including none at all. A PCOM and SCOM may appear on the same input line. Once a PCOM has been encountered, SCOMs associated with it are expected until the **END** subcommand is encountered. All input commands are processed until the **STOP** primary command is encountered. If the end-of-file is detected on the Job Input File prior to finding the **STOP** command, the **STOP** command will be internally generated and the program will end gracefully.

There are two options for processing input commands. These are the batch and interactive options. The batch option is the default option for processing commands from a batch input file. The **JOB** command allows the user to switch to interactive style processing of batch file commands. The interactive option is always active if the user uses the **TERM** command to activate interactive input from the terminal keyboard.

Batch Option Command Structure

Batch-style processing requires that primary commands always begin in column 1 of an input line. Subcommands may begin in any column other than column 1. For this option, using an **END** subcommand to terminate input for a primary command group is optional. If another primary command is encountered before an **END** subcommand is found, the missing **END** subcommand is internally generated and processing of the next primary command proceeds.

Batch Option Example

```
C       comments lines can go here.  Note that only PCOM or comment can start in column 1
PCOM
        SCOM  ( data here )     /*  comment fields may be appended to command lines
        SCOM  ( data here )     /*  subcommands may start in column 2 or greater
        END                     /*  END subcommand is optional
PCOM SCOM  (data here )         /*  SCOM may appear on PCOM command line
C       When next PCOM is found before END is encountered
C               an END command is automatically generated
PCOM
        SCOM  ( data here )
        END
STOP
C  Job processing terminates when the STOP command is encountered
C  If eof (end-of-file) is reached without finding a STOP command,
C           a STOP is generated automatically.
eof
```

Interactive Option Command Structure

Interactive processing allows primary commands and subcommands to start in any column of an input line. This option requires that an **END** subcommand be used to terminate input for a primary command group. If another primary command is encountered before an **END** subcommand is found, PZFlex attempts to interpret it as a subcommand of the current primary command group and terminates because of an error.

Interactive Option Example

```
  C  Comment lines may be used the same as for batch processing
        PCOM                          /* primary commands may start in any column
SCOM  ( data here )                   /* subcommands may start in any column
        END                           /* END subcommand is required
    PCOM SCOM  (data here )
END                                   /* END subcommand is required
STOP                                  /* STOP command behavior is the same as batch option
```

FREE FORMAT SYNTAX

1. Only the first 200 columns of an input line are read.

2. Command lines whose parameter list would exceed 200 characters may be continued on subsequent lines by placing an "&" character at the end of each line which is to be continued. There is no limit to the number of continuation lines which can be used to form a single command input line.

3. Up to 300 input parameters may be placed on a single command line.

4. All input parameters on a command line must be separated by one or more blanks.

5. Input parameters must appear on the command line in the order specified by the command's definition.

6. An * (asterisk) character as an input parameter's value denotes a default value.

7. All input parameters left blank at the end of a command line are automatically assigned default values.

8. Blank lines are ignored.

9. A floating point input parameter begins with a +(plus), -(minus), .(decimal point), or numeric character (0-9). An integer input parameter begins with a +(plus), -(minus), or numeric character. A character string must either begin with an alphabetic character (A-Z) or be enclosed in single quotes (').

10. Character data inputs are a maximum of 20 characters long unless otherwise noted. No embedded blanks are allowed.

LIST OF **PZFlex** COMMANDS

PCOM	SCOM	input parameters
		Define local coordinate system axes
AXIS		
	FORM	*option*
	DEFN	*axisname axistype* [list of parameters vary]
	BUFR	*nsiz_defn*
		Bond different grid discretizations together
BOND		
	DEFN	*bondname nocheck*
	CORS	*ibegin iend jbegin jend kbegin kend*
	FINE	*ibegin iend jbegin jend kbegin kend*
	FUZZ	*fuzzfactor*
		Specify boundary conditions for model (new procedure)
BOUN		
	SIDE	*[iside] type option value(1) value(2) ... value(6)*
	DEFN	*boundaryname type option value(1) value(2) ... value(6)*
	DOF	*dof(1) dof(2) ... dof(6)*
	NODE	*ibegin iend jbegin jend kbegin kend (matname)*
	MSHP	*pathname arrayname*
	MSHP	MATR *matname*
	MSHP	*pathname asxx asyy aszz asxy asyz asxz nquad*
	PLOT	*option*
		Specify which secondary variables will be computed
CALC		
	AINTN	
	ARST	
	AVRG	*avrgname arrayname weightoption region ibeg iend jbeg jend kbeg kend*
	BWORK	*surfacename ibeg iend jbeg jend kbeg kend normal*
	CURL	
	DISP	*component component component*
	ECENT	*elmarray nodarray (RATE irate)*
	ENRG	*energyname (option) ibegin iend jbegin jend kbegin kend*
	ENRGINTG	*(rate) name etype rtype ibeg iend jbeg jend kbeg kend*
	EQEP	*options*
	ESTP	*option*
	FRAG	*fragname ibegin iend jbegin jend kbegin kend bodyfile nadloop*
	INTEXT	*dataname type filename ncolbegin ncolend nskip*
	INTG	*dataname intgname*
	LOSS	*option*
	MATPROP	*arrayname matprop*
	MAX	*dataname minname maxname*
	MAXE	

PCOM SCOM input parameters

Define circuit models for use in piezoelectric applications

CALC (cont.)

P2IN	
PRES	
PTEN	*elmtype (ilayer) tensor (compres) name (RATE irate) ibegin iend jbegin jend kbegin kend*
PTEN	HEX *tensor (compres) name (vector) (RATE irate) nhexbeg nhexend*
PTEN	TET *tensor (compres) name (vector) (RATE irate) ntetbeg ntetend*
PZENRG	*ioptk ioptd iopte ioptc*
RESULTANT	XYZ *x0 y0 z0 v11 v21 v31 v12 v22 v32 side*
RESULTANT	IJK *i1 j1 k1 i2 j2 k2 i3 j3 k3 side*
RESULTANT	GCON *n1 n2 n3 side*
ROTN	
SJ2P	
STRN	*component component component component component component*
STRS	
TIMER	*option*
TFRM	*option* (*datanames*) TO *axisname ibegin iend jbegin jend kbegin kend*
VELM	
VMAG	*vmagname xdataname ydataname (zdataname)*
VOLM	*volumename (histname) ibegin iend jbegin jend kbegin kend*
VOLMINTG	*(rate) name array ibeg iend jbeg jend kbeg kend*
VOLR	
VPHASE	*datanamefrom datanameto (fmin fmax nf vmin vmax nv)*

Define circuit models for use in piezoelectric applications

CIRC

DEFN	*circname*
ELEM	*type connect value(s)*
DSRC	*qnsrc qnref nodsrc nodref scale*
PRNT	

PCOM	SCOM	input parameters

Perform input, output, and other operations with Data Manager arrays

DATA		
	ABS	*datanameabs dataname1 dataname2 . . . datanamen*
	ADD	*datanamefrom scale datanameto ibegin iend jbegin jend kbegin kend*
	CDDO	*dataname filename ibegin iend jbegin jend kbegin kend*
	CLOS	*dataname*
	COPY	*datanamefrom datanameto*
	CPYG	*datanamefrom ibgfrom iincfrom jbgfrom jincfrom kbgfrom kincfrom &* *datanameto ibgto iendto jbgto jendto kbgto kendto*
	DIVIDE	*dataname_denom datamane_numer*
	EXPAND	*array arraynew*
	EXPONENT	*dataname exponent*
	FFT	*realarray imagarray direction inverse*
	FFTSHFT	*array1 (array2 array3 …arrayN) direction*
	FILE	*option filename (type)*
	FORM	*option format idtag*
	HIST	*dataname nvalues filename scaletime scaledata*
	IMPORT	THRM *filename matname*
	IN	*dataname*
	INTERP	*array arraynew*
	INTEXT	*dataname type filename ncolbegin ncolend nskip*
	LOG	*dataname logdataname maxvalue*
	MATH	*arraynew* = [algebraic expression involving arrays]
	MGR	DIM *dataname ndimension ndim1 (ndim2) (ndim3)*
	MGR	MAP *dataname itype mapname*
	MIRR	*array arraynew axis option (sidopt) ibeg iend jbeg jend kbeg kend*
	MULTIPLY	*dataname1 dataname2*
	OPEN	*dataname ndimension ndimi ndimj ndimk datatype*
	OUT	*dataname*
	OUT0	*dataname*
	OUT1	*dataname*
	PHAD	*ampfrom phasfrom scale ampto phasto ibegin iend jbegin jend kbegin kend*
	POLR	*real imag option amp phas ibegin iend jbegin jend kbegin kend*
	REMAP	*pathname source_array destination_array axis nsample &* *ibegin iend jbegin jend kbegin kend*
	SCAL	*dataname scalevalue ibegin iend jbegin jend kbegin kend*
	TABL	*dataname idim jdim*
	VPHASE	*datanamefrom datanameto*

Select a dynamic relaxation analysis

DRLX	*relaxfreq*

Round-trip pulse echo option

ECHO		
	NODE	*ibegin iend jbegin jend kbegin kend iinc jinc kinc*
	RCRD	*xcrdrefl ycrdrefl zcrdrefl option*
	PULS	*pulse_duration quad_order stor_opt*
	SYMM	*symmetry_condition*

PCOM	SCOM	input parameters
		Execute a dynamic or static analysis
EXEC		*ntime endtime (static_option) print_option*
EXEC		CYCL *ncycle print_option*
EXEC		RINGDOWN (array) *ntimering nstavg stoptol ibeg iend jbeg jend kbeg kend offset*
EXEC		RINGDOWN electrode *eoptn ename ntimering nstavg stoptol offset*
		Save information for Fourier extrpolation including beam patterns
EXTR		
	REF	*refoption xref yref zref*
	DEFN	*extoption*
	DRIV	*driv_array*
	NODE	*ibegin iend jbegin jend kbegin kend iinc jinc kinc*
	TIME	*timebegin irate*
		Freefield boundary file processing, for either input or output
FFLD		
	FILE	*filename*
	IN	*axisname timeshift fuzzfactor option*
	OUT	*ibeginvel iendvel jbeginvel jendvel kbeginvel kendvel* & *ibeginstr iendstr jbeginstr j endstr kbeginstr kendstr*
	RATE	*irate_out*
	CORD	*active_coord(s)*
		Implement one way flow convection
FLOW		
	MATR	*mat1 mat2 mat3 … matn*
		Define the form of the callable function
FUNC		
	NAME	*namfnc*
	ASTEP	*amplitude2 steptime amplitude tdelay type exponent band amplitude3 steptime2*
	BLAK	*centerfrequency amplitude tdelay*
	CHIRP	*amplitude timeshift frq1 frq2 duration rampup rampdown*
	GAUS	*width amplitude tdelay*
	HIST	*histname scal adopt tdelay*
	SINE	*frequency amplitude phaseshift nperiod valuadd rampcycle tdelay*
	STEP	*amplitude2 steptime amplitude1 tdelay*
	TDELAY	*option*
	WVLT	*frequency amplitude gamma nu tdelay*

PCOM SCOM input parameters

Define a set of gage points in continuum elements

GAGE		
	LOC	*label array xgage ygage zgage*
	LINE	*label array xbeg xend ybeg yend (zbeg zend) npoints*
	BLOK	*label array axis xbeg xend ybeg yend (zbeg zend) nx ny nz*
	CYLN	*label array axis zbeg zend xcent ycent radbg radend thetabeg thetaend nr ntheta nz*
	SPHR	*label array xcent ycent zcent radbg radend thetabeg thetaend phibeg phiend nr ntheta nphi*
	END	

Define the coordinates of all grid nodes

GEOM		
	XCRD	*xbegin xend ibegin iend ratio*
	YCRD	*ybegin yend jbegin jend ratio*
	ZCRD	*zbegin zend kbegin kend ratio*
	KEYPNT	*nikey njkey nkkey*
	BEND	*axisname neutral_x neutral_y neutral_z ibgin iend jbgin jend kbgin kend*
	FILE	*infilename*
	SKEW	*(form) ibegin iend jbegin jend kbegin kend*
	PRNT	
	OUT	*form*

Glue pieces of grid together

GLUE		
	DEFN	*gluename keyword option*
	MASTER	*(option) ibegin iend jbegin jend kbegin kend*
	MASTER	*gcoption n1 n2 (n3 n4 n5 n6 n7 n8)*
	SLAVE	*(option) ibegin iend jbegin jend kbegin kend*
	SLAVE	*gcoption n1 n2 (n3 n4 n5 n6 n7 n8)*

Specify the number of grid nodes and type of problem

GRID	*igrid (jgrid) (kgrid) constraint*

PCOM SCOM input parameters

Color graphics display of the model and field variables

GRPH

ACTV	*type option*
ARROW	*option scale nspace ibegin iend jbegin jend kbegin kend*
BBOX	*type (xmin xmax ymin ymax zmin zmax)*
BLOK	*ibegin iend jbegin jend kbegin kend*
CLER	*type*
CLOS	
COLR	*(option) ival tred tgreen tblue hred hgreen hblue*
COMB	*(option) newname oldname1 oldname2 ... oldnamen*
DISP	*xscale yscale zscale*
DRAW	NODE *ibegin iend jbegin jend kbegin kend*
DRAW	CORD *xbegin ybegin zbegin xend yend zend*
DUPL	*xshift yshift zshift*
EDGE	
EYE	*xvalue yvalue zvalue*
FRAME	*x-space y-space z-space*
IMAG	
INVR	
LINE	*option*
MAP	*materialname colorindex*
MIRR	*axis option dataoption*
NODE	*option*
NVEW	*nview*
PAN	*pan_option zoom_value (all_option) (save_option)*
PLOT	(*ijk_option*) MATR (BLOK *blockname*)
PLOT	[*datan*] (BLOK *blockname*) (RANG *minvalue maxvalue*)
PLOT	SHAP *numshape* (*shapdata*) *phaseshift*
PLOT	FLXHST *irecordn*
PS	*option*
REVR	*iaxis option*
SET	*option parameters*
SIZE	*xsize ysize xbegin xend ybegin yend nxdiv nydiv*
SWAP	*option*
TIME	*itimestep time*
TTL	*ititle*

<----------200-character title information for line = *ititle*---------->

VERT	*xvalue yvalue zvalue*
VIEW	*iview*
VPNT	*xvalue yvalue zvalue*
WNDO	(option1) (option2) *xbegin xend ybegin yend zbegin zend*
WRIT	(*options*) *filename*
WSIZ	*width height*
ZOOM	*zoom_value (all_option) (save_option)*

PCOM SCOM input parameters

Interface smoothing

GSMU

WNDO	*ibeg iend jbeg jend kbeg kend*
PARM	*bandwidth niter*
CALC	
LIMIT	***tlimit***
END	

Specify computational parameters specific to the thermal solver

HEAT

WNDO	*ibeg iend jbeg jend kbeg kend*
SLVR	*solvertype alpha droptol*
NLIN	*nstprfr maxits nitrfr rtol relax prnoption*
CUPL	*option*
END	

Switch between batch and interactive command interpretation options

JOB option

Activates large deformation option for skewed partition

LDEF option

Defines 2-noded line elements

LINE

TYPE	*linetype*
PROP	<--- input parameters vary depending on *linetype* -->
CURV	*curvname (formoption) (hardoption) disp1 forc1 & disp2 forc2 ... dispn forcn*
LDEF	*option*
ELEM	*linename ibegin iend jbegin jend kbegin kend*
ORFC	*n1 n2 area temperature gas_cons gamma*
SNGL	*linename i1 j1 k1 i2 j2 k2*
ITFC	*linename ib1 ie1 jb1 je1 kb1 ke1 ib2 ie2 jb2 je2 kb2 ke2 xr yf zr refopt*
PRNT	
INFO	*option1 option2*

Specify magnetostrictive processing information for the model

MAGN

WNDO	*ibegin iend jbegin jend kbegin kend*
DEFN	*coilname*
NODE	*turns ibegin iend jbegin jend kbegin kend*
PMAG	*mx my ibegin iend jbegin jend kbegin kend*
BC	*coilname option histname scalehist shifthist*
CALC	*option*
HRGL	*hourglass*
SIDE	*iside type option value(1) value(2) ... value(6)*
CONN	*coilname circuitname sourceoption histname scalehist shifthist*
SLVR	*option*

PCOM SCOM input parameters

Assign concentrated masses to grid nodes

MASS

ADD	*masstype value i-index j-index k-index*
SET	*masstype value i-index j-index k-index*
GCON	*option masstype nodegc value*
PRNT	

Define continuum material properties

MATR

TYPE	*mattype*
PROP	*matname density bulk shear hourglas damplinear dampquad* [see specific model types for parameters and valid codes]
HRGL	*matname keyword(s)*
IMPD	*matname cp cs*
DAMP	*matname volrate*
WVSP	*option*
VDMP	*matname centfreq option dilatation shear frequency expv distance exps*
SDMP	*matname centfreq option dilatation shear frequency expv distance exps*
MDMP	*matname centfreq option damping wavetype frequency expv distance exps*
RDMP	*matname centfreq option dilatation shear frequency expv distance exps*
SLOSS	*matname mechfactr dielfactr piezfactr*
FRNG	*frqmin frqmax wtshear*
PRNT	*option matname frqmin frqmax frqinc (filename)*
NWTN	*matname volumetric shear*
AXIS	*matname axisname*
AFRC	*matname fractabsorb*
PIEZ	*matname irow1 jcol1 value1 irow2 jcol2 value2 ... irown jcoln valuen*
ELEC	*matname epsilonx epsilony epsilonz*
STOR	*option*
PERM	*matname (option) permx permy permz*
MGST	*matname (option) irow2 jcol1 value1 irow1 jcol2 value2 irown jcoln value n*
MGST	*matname psat kappa epsilon0 q111 q211*
RATE	*matname beta expnt edot1 factor1 edot2 factor2 … edotn factorn*
THRM	*matname C kx' ky' kz' axisnam hgls cb wb TB*
COPY	*copyfrom copyto*
CURV	*dataname nvalues filename scalex scaley shiftx (option)*
PARM	*store_option*

Define 4-noded membrane elements for 2D or 3D problems

MBRN

PROP	*membname rho emodulus poisson sigyield thickness tensyield hourglas*
ELEM	*membname ibegin iend jbegin jend kbegin kend*
SNGL	*membname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
PRNT	*option*

PCOM SCOM input parameters

Allocate RAM memory available to the job

MEM *memory_numeric memory_character option*

Prints the Data Manager structure and check its consistency

MGR

Define special model characteristics of continuum elements

MODL

SLIP	*modelname cohesion1 tangent1 cohesion2 tangent2*
REBR	*modelname emodulus sigyield ratio1 ratio2 ratio3 viscous vshape*
DBND	*modelname slipname tcutoff*
FLUD	*modelname phi0 bulks n e tenf umax sat*
PCEL	*modelname presname xscale yscale*
VCTR	*vectorname vx1 vy1 vz1 vx2 vy2 vz2*
ELEM	*modelname vectorname ibegin iend jbegin jend kbegin kend matnam*
PRNT	*option iaxis ibegin iend jbegin jend kbegin kend*
END	

Modifies the model after execution has begun (large deformation only)

MODS

OPTN	*elmopt erodopt erodval plodopt erodmat*
EROD	*ibegin iend jbegin jend kbegin kend*
PLOD	*option ibegin iend jbegin jend kbegin kend*
PRNT	*print_option*
GRUP	*grpname*
GEND	
TINC	*grpname inctime*
STOP_NODES	*option disprange*
TIMCHK	*option*

Control shared-memory parallel processing

MP

OMP	*maxthreads dynoption*

PCOM SCOM input parameters

Specify piezoelectric processing information for the model

PIEZ

WNDO	*ibegin iend jbegin jend kbegin kend*
DEFN	*electrodename*
NODE	*iend jbegin jend kbegin kend*
NOD2	*mat1 mat2 ibegin iend jbegin jend kbegin kend*
NOD3	*objectname side ibegin iend jbegin jend kbegin*
BC	*electrodename option histname scalehist shifthist*
CONN	*electrodename circuitname sourcetype histname scalehist shifthist*
SLVR	*option*
AXIG	*arcangle factor ibegin iend jbegin jend*
SIDE	*iside type option value(1) value(2) … value(6)*
CNVR	*convergence*
CALC	*option*
IMPD	*frequency scalecurrent scalevoltage*
HRGL	*hourglass*
TIME	*time*
ESTA	*nskip*
EXEC	*numitr*
PRNT	*option*
NONL	*rtol maxits*
CUPL	*copt*
ECON	*electrode1 electrode2 type value(s)*

Pressure load option to provide user with complete control over pressure loading

PLOD

PDEF	*plodname presname presscale timeshift timescale*
SWEP	*plodname presname presscale timeshift rateswep vxswep &* *vyswep vzswep xrefswep yrefswep zrefswep*
SPOT	*spotname sx sy sz*
VCTR	*vectorname vx vy vz*
MATR	*mname material option*
MTTL	*vectorname vx vy vz*
SDEF	*plodname direcname* (LINE *width*) *i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
SDF2	*plodname direcname matname1 matnam2 ibegin iend jbegin jend kbegin kend*
SDF2	*plodname direcname matname1 side iside ibegin iend jbegin jend kbegin kend*
CYLN	*plodname direcname (axisname) iaxis cbegin cend center1 center2 radius thetabeg* *thetaend &* *ibegin iend jbegin jend kbegin kend*
SPHR	*plodname direcname xcent ycent zcent radius thetabeg thetaend phibegin phiend &* *ibegin iend jbegin jend kbegin kend*
SNGL	*plodname direcname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
GCON	*plodname direcname nodegc1 nodegc2 nodegc3 nodegc4*
LDEF	*option*
CHEK	*overlapcheck voidcheck*
DACT	*plodname option*
PRNT	*option*

PCOM SCOM input parameters

Request printed output during an analysis. Also saves time history results to disk.

POUT		
	FORM	*format idtag*
	RATE	*iratehist iratesnap*
	HIST	*datanames ibegin iend iinc jbegin jend jinc kbegin kend kinc*
	HISTNAME	OBJECT_TYPE *option object_name*
	PLOT	option
	SNAP	
PRNT		<----------standard **PRNT** primary command---------->
SHOW		<----------standard **SHOW** primary command---------->

Process the predefined model to prepare it for execution

PRCS

Prints values for arrays stored in the Data Manager

PRNT *dataname option ibegin iend jbegin jend kbegin kend*

Revise the mesh during analysis

REGRID		
	FILL	(*axisname*) *fillindex ibegin iend jbegin jend kbegin kend icomp*

Restart a previous job or suppress writing of restart files

REST (parameter list is dependent on desired action)

Defines rigid substructures

RIGD		
	DEFN	*rigidname xtmass ytmass ztmass xrmass yrmass zrmass xyrmass yzrmass xzrmass*
	CG	*xcg ycg zcg*
	NODE	*ibegin iend jbegin jend kbegin kend (matname) (matnam2*
	GCON	*nodegc1 nodegc2 nodegc3 ... nodegcn*
	FIX	*ifix1 ifix2 ifix3 ifix4 ifix5 ifix6*
	FREE	*dof(1) dof(2) ... dof(6)*
	BC	*bctype histname scalx scaly scalz scalrx scalry scalrz*
	SLAV	*inode jnode knode*
	CHNG	*rigidname option* [parameters]
	COMB	*maxloop*

Set the value of array variables that are stored in the Data Manager

SET *dataname value (add) ibegin iend jbegin jend kbegin kend*

PCOM	SCOM	input parameters

Extracts steady-state deformation shapes from linear transient analysis

SHAP		
	BASE	*arrayname ibase jbase kbase*
	CNVRT	*option*
	DATA	*arrayname*
	FREQ	*frequency*
	NODE	*ibegin iend jbegin jend kbegin kend*
	TWND	*[timewindow] type option fraction*

Define shell elements for 2D or 3D problems

SHEL		
	TYPE	*proptype*
	PROP	*pname rho emodulus poisson sigy hmod siglim gfac edot(1) sig(1)* & *edot(2) sig(2) ...edot(n) sig(n)*
	PROP	*pname gfac*
	PROP	*PNAME E1 E2 v12 G12 G23 G31 sigyt1 sigyt2 tauy12 tauy23* & *tauy12 tauy23 tauy31 gfac (ncr sigyc1 sigyc2)*
	THEX	*mname alphax alphay alphaz*
	AVIS	*mname* MATR *alinear aquadratic cmpopt strnrat*
	AVIS	*mname* CRIT *frequency fracdmp ellength*
	INFO	*option [parameters]*
	LAYR	*(option) lname thick nlayer pname(1) ...pname(nlayer)* & *hginpln hgnorm hgrot rotfac*
	ELEM	*lname ibegin iend jbegin jend kbegin kend (mat1) (mat2)*
	SNGL	*lname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
	GCON	*lname nodegc1 nodegc2 nodegc3 nodegc4 (groupnames)*
	DEL	*layrname ibeg iend jbeg jend kbeg kend*
	THIK	*lname fractiona1 fraction2 … fraction*
	OFFS	*lname zoffset*
	ANGLE	*lname option angle1 angle2 ... anglen*
	BUFR	*nsiz_elem*
	LABL	*labelname ibegin iend jbegin jend kbegin kend touch_option*
	PRNT	*option*

Print a normalized map of value amplitudes for arrays stored in Data Manager

SHOW	*dataname iaxis ibegin iend jbegin jend kbegin kend normvalue*

PCOM SCOM input parameters

Assigns material properties to each continuum element in the model

SITE

CYLN	*matname (iaxis) cbegin cend center1 center2 radbegin radend*
BLOK	*matname option xbegin xend ybegin yend zbegin zend &*
	ibegin iend jbegin jend kbegin kend
ELIP	*matname xcenter ycenter zcenter axisname a b c*
FLIT	*matname axisname 0x 0y 0z a b c shape ibegin iend jbegin jend kbegin kend*
LOCK	*matnam1 matnam2 matnam3…*
POLY	*matname (axis) ibegin iend jbegin jend kbegin kend zbeg zend &*
	x1 y1 x2 y2 x3 y3 … xN yN
REGN	*matname ibegin iend jbegin jend kbegin kend*
REGNCOPY	*(option) (option) (option) ibegf iendf jbegf jendf kbegf &*
	Kendf ibegt iendt jbegt jendt kbegt kendt
REGNDUPL	*ibegf iendf jbegf jendf (kbegf)(kendf) dupdir ncopy invertopt*
SPHR	*matname xcenter ycenter zcenter radius*
SWAP	*matname matnew ibegin iend jbegin jend kbegin kend*
TABL	*option filename scale xshift yshift zshift ibegin iend jbegin jend kbegin kend resize*
TETR	*matname axisname x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4 &*
	Ibegin iend jbegin jend kbegin kend
TORUS	*matname (axis) xcenter ycenter zcenter rbig rsmall thetabeg thetaend phibeg phiend &*
	ibegin iend jbegin jend kbegin kend
MMAP	*format filename ibeg iend jbeg jend kbeg kend nxbin nybin nzbin &*
	axisname scalc scalx scaly scalz midofs
ARAY	*arrayname matname vmin vmax ibegin iend jbegin jend kbegin kend*
OBJT	*matname objname*
PRNT	*iaxis ibegin iend jbegin jend kbegin kend*

Stop processing input commands

STOP

Controls routing of standard input or output to the terminal

TERM *inoption outoption*

Sets the time parameters for the model

TIME *timestep timebegin timefactor ratiomin timemin*

Specifies the job title and id

TITL *<--id-->* *<----------title---------->*

Calls user-written routine to perform user-specified purpose

USER -parameters are dependent on the user written routine-

PCOM SCOM input parameters

Specifies the computational window for the model to conserve computer resources

WNDO *ibegin iend jbegin jend kbegin kend*

Exchanges file with foreign codes

XFIL

FORM	*codeoption*
FILE	*filename*
WRIT	*writeoption (tag)*
SYMM	*axis_normal axis_coord fuzz_factor*

Define processing zones for the model

ZONE *iratio ibegin iend jbegin jend kbegin kend zonestep*

AXIS (axis) Input Command

Purpose: To define local coordinate systems relative to the global Cartesian system. These new local systems can be accessed by other code options in order to facilitate transforms between global and local systems.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any other command which references the axis definitions.

PCOM SCOM input parameters

AXIS

FORM *option*

<<option = CORD>>
DEFN *axisname axistype x0 y0 z0 x1 y1 z1 x2 y2 z2*
<<option = VCTR>>
DEFN *axisname axistype x0 y0 z0 vx1 vy1 vz1 vx2 vy2 vz2*
<<option = ANGL>>
DEFN *axisname axistype x0 y0 z0 rotateX rotateY rotateZ*

BUFR *nsiz_defn*
END

FORM Subcommand
AXIS Input Group

AXIS-FORM

Purpose: To change the form of the local system definition on the **DEFN** subcommand.

Use Is: Optional. If not input, *option*=VCTR is assumed.

Multiple Use: Allowed.

Order Dependence within **AXIS** Group: A **FORM** subcommand must precede a **DEFN** subcommand.

PCOM SCOM input parameters

FORM *option*

option= (C) Local axis definition option.
Any of: VCTR, CORD or ANGL

If *option*=VCTR, then the **DEFN** command defines the origin and two vector directions to define the local system.

If *option* = CORD, then the **DEFN** command defines the origin and two spatial locations that define will be used to compute the vectors defining the local system axes.

If *option* = ANGL, then the **DEFN** command defines the origin and three rotation angles about the global axes that will transform a global system into the desired local system.

Default is VCTR.

DEFN (definition) Subcommand
AXIS Input Group **AXIS-DEFN**

Purpose: To define a local coordinate system by defining a local Cartesian system (x', y', z') which orients the desired coordinate system relative to the global Cartesian space as shown in the attached figure.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **AXIS** Group: A **DEFN** subcommand must follow a **FORM** subcommand.

PCOM SCOM input parameters

<< if *option* = CORD -(set by **FORM** subcommand)>>
<< define the local system by three spatial coordinates >>

DEFN *axisname axistype x0 y0 z0 x1 y1 z1 x2 y2 z2*

axisname= (C) Name of this coordinate axis definition.
No default.

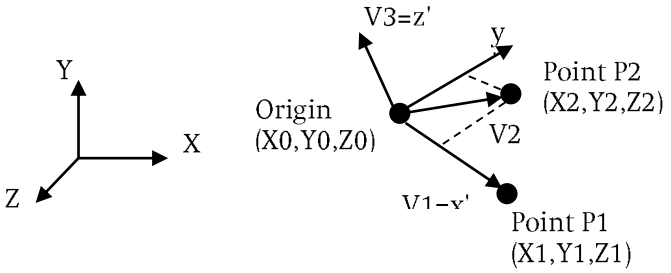
axistype= (C) Type of this coodinate system. (See Notes)
Any of: CART = Cartesian system
CAR2 =Cartesian system definition 2
CYLN =cylindrical system
CYL2 =cylindrical system definition 2
SPHR =spherical system
No default.

x0, y0, z0= (F) Global coordinates of the origin of this coordinate axes.

x1, y1, z1= (F) Defines the global coordinates of Point P1, this being a coordinate on the new x-axis, x' for **CART** or **CYL,** or on the new z-axis, z' for **CAR2** and **CYL2**. See Illustration overleaf.

x2 y2, z2= (F) Defines the global coordinates of Point P2**,** which is a coordinate on the new x-y plane for **CART** and **CYL,** or on the new x-z plane for **CAR2** and **CYL2**. See illustration overleaf.

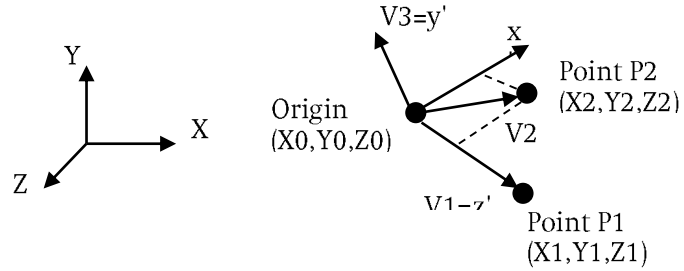
AXIS-DEFN



Local system axes

x' = V1
z' = V3 = V1 x **V2**
y' = z' x **x'**

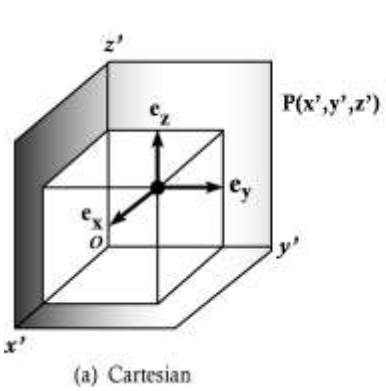
option = CORD *axistype* = CART, CYLN, or SPHR



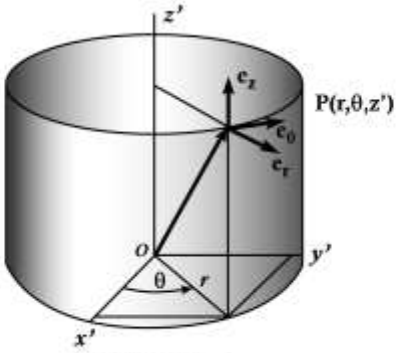
Local system axes

z' = V1
y' = V3 = V1 x **V2**
x' = y' x **z'**

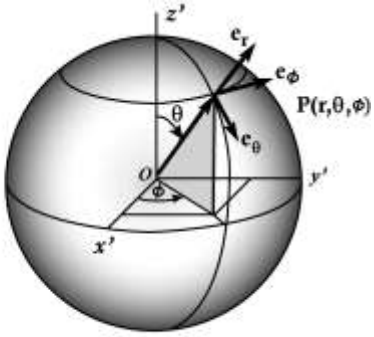
form = CORD *axistype* = CAR2 or CYL2



(a) Cartesian



(b) Cylindrical



(c) Spherical

$$r = \sqrt{(x')^2 + (y')^2}$$

$$\theta = \tan^{-1}\left(\frac{y'}{x'}\right)$$

$$r = \sqrt{(x')^2 + (y')^2 + (z')^2}$$

$$\theta = \cos^{-1}\left(\frac{z'}{r}\right)$$

$$\phi = \tan^{-1}\left(\frac{y'}{x'}\right)$$

AXIS-DEFN

PCOM SCOM input parameters

<< if *option* = VCTR -(set by **FORM** subcommand)>>
<< define the local system by one spatial location and two vectors >>

DEFN *axisname axistype x0 y0 z0 vx1 vy1 vz1 vx2 vy2 vz2*

axisname= (C) The name of this coordinate axis definition.
No default.

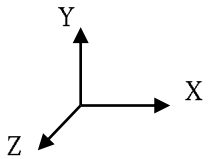
axistype= (C) The type of this coodinate system. (See Notes)
Any of: CART = Cartesian system
CAR2 =Cartesian system definition 2
CYLN =cylindrical system
CYL2 =cylindrical system definition 2
SPHR =spherical system
No default.

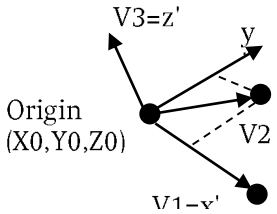
x0, y0, z0= (F) Global coordinates of the origin of this coordinate axes.

vx1, vy1, vz1= (F) Defines the x-,y-, and z-components of vector V1, which is a vector for the new x-axis, x' for **CART** or **CYL** or the vector for the new z-axis, z' for **CAR2** and **CYL2**. See illustration overleaf.

vx2 vy2, vz2= (F) Defines the x-,y-, and z-components of vector V2, which is a vector on the new x-y plane for **CART** and **CYL,** or a vector on the new x-z plane for **CAR2** and **CYL2**. See illustration overleaf.

AXIS-DEFN

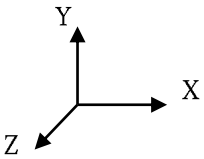


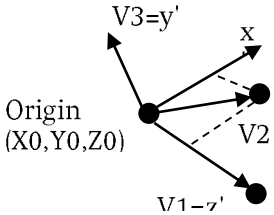


Local system axes

x' = V1
z' = V3 = V1 x **V2**
y' = z' x **x'**

option = VCTR *axistype* = CART, CYLN, or SPHR

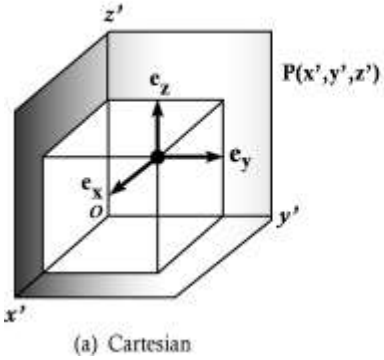




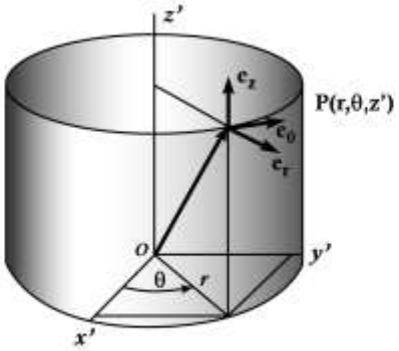
Local system axes

z' = V1
y' = V3 = V1 x **V2**
x' = y' x **z'**

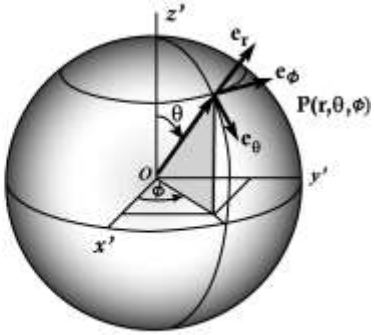
form = CORD *axistype* = CAR2 or CYL2



(a) Cartesian



(b) Cylindrical



(c) Spherical

$$r = \sqrt{(x')^2 + (y')^2}$$

$$\theta = \tan^{-1}\left(\frac{y'}{x'}\right)$$

$$r = \sqrt{(x')^2 + (y')^2 + (z')^2}$$

$$\theta = \cos^{-1}\left(\frac{z'}{r}\right)$$

$$\phi = \tan^{-1}\left(\frac{y'}{x'}\right)$$

AXIS-DEFN

PCOM SCOM input parameters

<< if *option* = ANGL-(set by **FORM** subcommand)>>
<< define the local system by one spatial location and three rotation angles >>

DEFN *axisname axistype x0 y0 z0 rotateX rotateY rotateZ*

axisname= (C) Name of this coordinate axis definition.
No default.

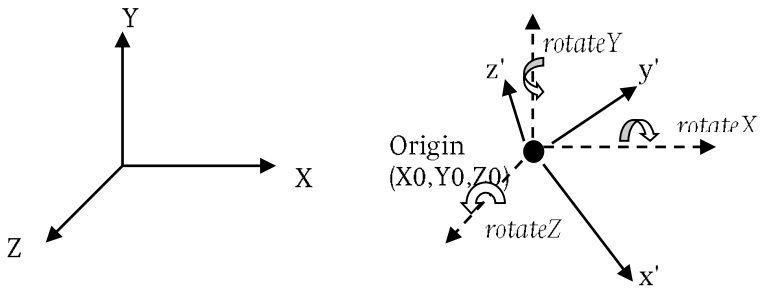
axistype= (C) Type of this coodinate system.
Any of: CART = Cartesian system
CYLN =cylindrical system
SPHR =spherical system
No default.

x0, y0, z0= (F) Global coordinates of the origin of this coordinate axes.

rotateX = (F) Number of degrees of rotation about the global X axis to align with the local system.
Default = 0.0

rotateY = (F) Number of degrees of rotation about the global Y axis to align with the local system.
Default = 0.0

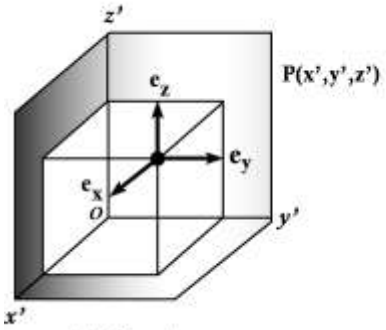
rotateZ = (F) Number of degrees of rotation about the global Z axis to align with the local system.
Default = 0.0



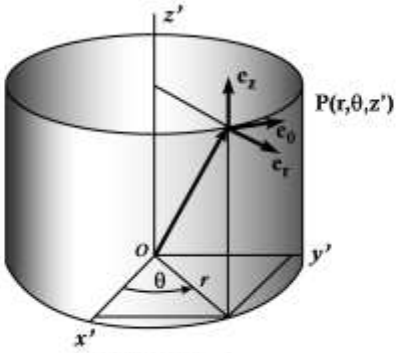
The local system is a composite created by applying first the ***rotatex***, then ***rotatey,*** and finally ***rotatez*** to the local system.

Rotation applied using the right-handed rule.

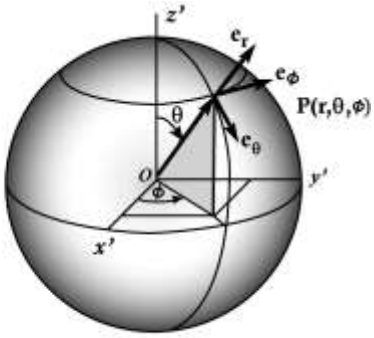
form = ANGL *axistype* = CART, CYLN or SPHR



(a) Cartesian



(b) Cylindrical



(c) Spherical

$$r = \sqrt{(x')^2 + (y')^2}$$

$$\theta = \tan^{-1}\left(\frac{y'}{x'}\right)$$

$$r = \sqrt{(x')^2 + (y')^2 + (z')^2}$$

$$\theta = \cos^{-1}\left(\frac{z'}{r}\right)$$

$$\phi = \tan^{-1}\left(\frac{y'}{x'}\right)$$

Orientation of local coordinate systems relative to global Cartesian system x, y, z

Notes: 1. For all cases except *axistype*=CYL2 and CAR2, vector V1 defines the the local x' axis. The z' axis is defined by V1 X V2. The y' axis is defined by z' X x'.

2. When *axistype*=CYL2 or CAR2, vector V1 defines the local z' axis. The y' axis is defined by V1 X V2. The x' axis is defined by y' X z'.

BUFR (buffer) Subcommand
AXIS Input Group **AXIS-BUFR**

Purpose: Change the incremental memory block size allocated to arrays created with AXIS commands. (NOTE: this command is typically unnecessary unless prohibitively large read-in times are experienced)

Use Is: Optional. If not input, default memory allocation is used.

Multiple Use: Allowed.

Order Dependence
Within **AXIS** Group: Must precede the DEFN subcommand.

PCOM	SCOM	input parameters
	BUFR	*nsiz_defn*

nsiz_defn = (I) The number of definitions created with DEFN subcommands to allocate space for at a time.
Default = 10. See Note 1.

Notes: 1. This option is not typically required. It should only be used if a model contains a very large number of axis definitions and the read-in time is long. The axis arrays are resized after read-in so it is not possible to over-allocate memory, but care should be taken not to enter a number larger than the memory space immediately available on the machine.

BOND Input Command

Purpose: To define surfaces of the grid that are bonded together. The surfaces may have different discretizations; the discretization of the more finely discretized surface, however, must be an integer multiple of the discretization of the coarser surface. Nodes on the coarse surface are master nodes, nodes on the fine surface are slave nodes.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow **GRID** command and precede **PRCS** command.

PCOM SCOM input parameters

BOND

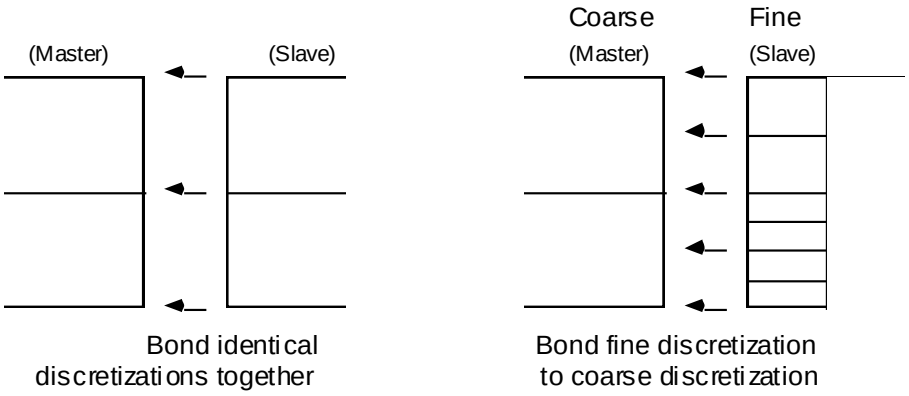
DEFN *bondname nocheck*

CORS *ibegin iend jbegin jend kbegin kend*

FINE *ibegin iend jbegin jend kbegin kend*

FUZZ *fuzzfactor*

END



Note: When kinematic boundary conditions are applied to bonded interface nodes, they must be applied to the coarse (master) nodes.

DEFN (definition) Subcommand
BOND Input Group **BOND-DEFN**

Purpose: To begin the definition of a new bonded interface.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
within **BOND** Group:Must precede the **CORS** and **FINE** subcommands.

PCOM SCOM input parameters

DEFN *bondname (nocheck) (pinopt)*

bondname = (C) Unique name associated with the bonded interface being defined.

nocheck = (C) If set to NCHK, option to override the checking of coordinates and order of bonding for this bonded set. In general, this override option should not be used.
Using this option requires that each coarse element has the same number of fine elements bonded to it, i.e., variable discretization cannot be evaluated without checking the coordinates.
Default: all internal code checks are made for this bonded set.

pinopt = (C) If set to PIN, only the translational degrees of freedom are bonded. Default: all degrees of freedom are bonded.

CORS (coarse surface) Subcommand
BOND Input Group **BOND-CORS**

Purpose: To define the more coarsely discretized surface being bonded.

Use Is: Required.

Multiple Use: Allowed. Only one occurrence, however, is allowed for each **DEFN** subcommand.

Order Dependence within **BOND** Group: To be used after the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	CORS	*ibegin iend jbegin jend kbegin kend*

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the coarsely discretized surface region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the coarsely discretized surface region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the coarsely discretized surface region.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. As the bonded surface is assumed to be a plane in IJK space, *ibegin* = *iend*, *jbegin* = *jend*, or *kbegin* = *kend*.
2. The coordinates of the beginning and ending nodes of the coarse surface should be coincident with the beginning and ending coordinates of the associated fine surface.
3. Coarse side nodes must not have been defined as fine side nodes for a previously declared bonded interface.

FINE (fine surface) Subcommand
BOND Input Group **BOND-FINE**

Purpose: To define the more finely discretized surface being bonded.

Use Is: Required.

Multiple Use: Allowed. Only one occurrence, however, is allowed for each **DEFN** subcommand.

Order Dependence within **BOND** Group: To be used after the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	FINE	*ibegin iend jbegin jend kbegin kend*

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the finely discretized surface region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the finely discretized surface region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the finely discretized surface region.
Default: *kbegin* = 1, *kend* = maximum K

Notes:
1. As the bonded surface is assumed to be a plane in IJK space, *ibegin* = *iend*, *jbegin* = *jend*, or *kbegin* = *kend*.
2. The coordinates of the beginning and ending nodes of the fine surface should be coincident with the beginning and ending coordinates of the associated coarse surface.
3. The bonding algorithm assumes that the fine side discretization is equal spaced. The nodal coordinates of the fine side, however, are not actually checked to ensure this. If the fine side spacing is not equal, the bonding logic is less accurate.

FUZZ (fuzz factor) Subcommand
BOND Input Group **BOND-FUZZ**

Purpose: To define the tolerance factor for evaluating whether or not the bonded faces are properly located relative to each other.

Use Is: Optional. If not input, the fuzz factor defaults to .00001.

Multiple Use: Allowed.

Order Dependence within **BOND** Group: The tolerance set by this subcommand applies only to bonded surfaces that follow this command.

PCOM	SCOM	input parameters
	FUZZ	*fuzzfactor*

fuzzfactor = (F) The value of the fuzz factor.
No default.

Note: An error is generated for any bond definition for which the coordinates of the two surfaces being bonded do not coincide within an acceptable tolerance distance. This tolerance distance is defined as the distance between the diagonal nodes of the coarse side multiplied by *fuzzfactor*.

BOUN (boundary) Input Command

Purpose: To assign boundary conditions to the grid boundary surfaces. This option supersedes the old **BDRY** command group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow **GRID** command and precede **PRCS** command.

PCOM SCOM input parameters

BOUN

SIDE [*iside*] *type option value(1) value(2) ... value(6)*

DEFN *boundaryname type option value(1) value(2) ... value(6)*

DOF *dof(1) dof(2) ... dof(6)*

NODE *ibegin iend jbegin jend kbegin kend (matname)*

MSHP *pathname arrayname*

MSHP MATR *matname*

MSHP *pathname asxx asyy aszz asxy asyz asxz nquad*

PLOT *option*

END

SIDE Subcommand
BOUN (boundary) Input Group **BOUN-SIDE**

Purpose: To provide a shorthand way to assign boundary conditions to an entire side of a grid.

Use Is: Optional.

Multiple Use: Allowed.

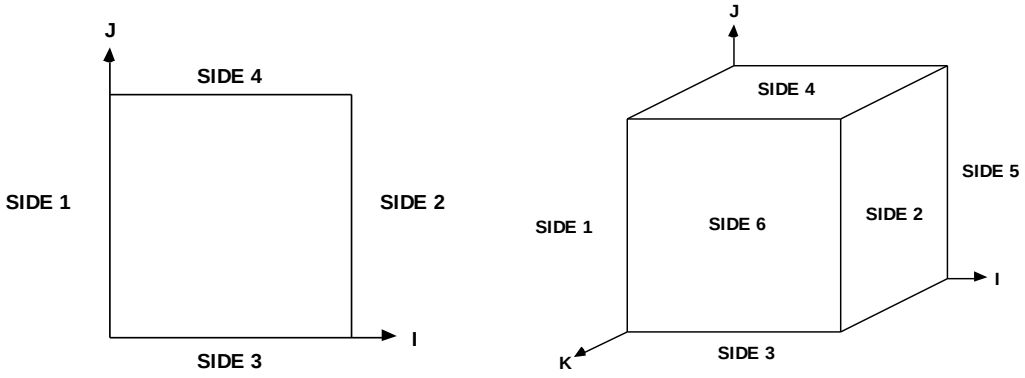
Order Dependence within **BOUN** Group: None. Note that boundary conditions are applied in the order in which they are input.

PCOM SCOM input parameters

SIDE [*iside*] *type option (option2) value(1) value(2) ... value(6)*

iside = (X) The "side identifier" specifying which side of the mesh to apply the specified boundary condition on. Side identifiers are shown here and are related to the mesh as shown in the 2D and 3D model figures below.

Side	Equivalent side identifiers		
side 1	1	XMIN	IMIN
side 2	2	XMAX	IMAX
side 3	3	YMIN	JMIN
side 4	4	YMAX	JMAX
side 5	5	ZMIN	KMIN
side 6	6	ZMAX	KMAX



2D models 3D models

Side identifier relationship to grid indices orientation

type = (C) Type of boundary conditions to be applied on side *iside*. No default.
Any of:
- FIXD = fixed boundary
- SYMM = symmetry boundary
- ASYM = antisymmetry boundary
- ABSR = absorbing boundary
- VEL = prescribed velocity boundary
- FFLD = freefield boundary (velocity & absorber)
- ISLN = soil island boundary
- USER = user-written subroutine USRBOU is used to apply boundary conditions
- TMPR = prescribed temperature (Thermal)
- FLUX = prescribed heat flux (Thermal)
- RADN = linearized surface radiation condition (thermal)
- INFN = infinite element (thermal)
- RAD4 = nonlinear surface radiation condition (thermal)
- CNVC = surface convection condition (thermal)
- IMPD = impedance boundary (with optional spring)

option= (C) Option for the type of boundary condition specified. Valid choices for *option* depend on the choice of *type*.

option2= (C) Option for the type of boundary condition specified. Valid choices for *option2* depend on the choice of *type*.

Boundary type	options
FIXD	none
SYMM and ASYM Default = ALL	TRAN- applies only to translational dofs ROT- applies only to rotational dofs ALL - applies to all dofs
ABSR and FFLD Default = NEW	STND- first order, less accurate for low CFL AUTO- Automatic optimization of boundary for solid and acoustic materials GRAZ- Grazing incidence, see note 3 below UPDT- updates absorber geometry for large deformation using STND logic UPDG- updates absorber geometry for large deformation using GRAZ option
VEL No default	FUNC - uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified using the **DATA HIST** option
ISLN No default	UPST - upstream boundary of soil island DNST - downstream boundary of soil island BOTM - bottom of soil island
TMPR No default	FUNC – uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option
RADN No default	FUNC – uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option UNIT - Constant
RAD4 No default	FUNC – uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option UNIT - Constant
CNVC No default	FUNC – uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option UNIT - Constant Option2 should be set to UNIT for a constant or the name of a digitized curve previously specified via the **DATA HIST** option.
FLUX No default	FUNC – uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option
INFN	RADL – element diverges radially NORM – element projects normally
USER	User-defined

value(n)= (F) The *value* inputs are applicable only to the VEL, SYMM, ASYM, RADN and FLUX boundary types.

For VEL boundaries, these values are the scale factors for each global degree of freedom, 1 through 6 (x,y,z,rx,ry,rz), respectively. The prescribed velocity for each dof of a node is the value of the prescribed time history, specified by *option*, scaled by the *value* input for that dof. The DOF subcommand can be used to control which degrees of freedom the VEL boundary applies to.

For SYMM and ASYM boundaries, typically the *value* inputs are not used. If the node or nodes for which a SYMM or ASYM boundary has been specified do not define a plane, however, then the first three *value* inputs may be used to define a vector that is normal to the desired symmetry plane.

For TMPR boundaries, *value(1)* is the scale factor for the temperature. The prescribed temperature at each node is the value of the prescribed time history, specified by *option*, scaled by *value(1).*

For RADN boundaries the heat flux is given by the relation: *value(1)** T — *value(2)**T_{inf}. Typically *value(2)=value(1)*. The prescribed temperature-at-infinity is the value of the time history, specified by *option.*

For RAD4 boundaries, the heat flux is given by the relation: value(1)* (T^4 —T_{inf}^4). The prescribed temperature-at-infinity is the value of the time history, specified by *option*.

For CNVC boundaries, the heat flux is given by the relation: value(1)*h(T_{avg})*(T —T_{inf})^{expont}. The prescribed temperature-at-infinity is the value of the time history, specified by *option*. h(T_{avg}) is defined by the digitized curve specified as *option2*. Here T_{avg} = 0.5* (T + T_{inf}). *expont* is specified as value(2).

For FLUX boundaries, *value(1)* is the scale factor for the heat flux. The prescribed flux at each node is the value of the prescribed time history, specified by *option*, scaled by *value(1).*

For INFN boundaries, *value(1), value(2), value(3)* are the (x,y,z) coordinates of the reference point.

For IMPD boundaries, the values are *density, p- wavespeed, s-wavespeed, (normal spring), (tangent spring.* See note 7

Notes:
1. The **PLOD** command can be used to apply general pressure boundary conditions on the model
2. Each boundary defined with a **SIDE** subcommand is internally assigned a unique 4-character name. These names are FLX1, FLX2, ..., FLX6 for sides 1 through 6, respectively. If the **DEFN** subcommand is used to define other boundaries, they should be assigned names that do not conflict with those provided by the **SIDE** subcommand. **PLOD** command can be used to apply general pressure boundary conditions on the model.
3. The GRAZ option adjusts the absorber logic so that it is accurate for perfectly normal or perfectly grazing incidence. The accuracy with this option over the range of other incidence angles is poorer overall than with the default option. It should be considered only when grazing incidence waves are the primary waves impinging upon the boundary. STND is less accurate than NEW for low CFL numbers, but could be more stable in some situations.
4. For TMPR, FLUX, RADN, RAD4, CNVC, or INFN boundaries, side refers to the thermal window rather than the entire grid. As with mechanical boundaries, a unique 4-character name is assigned to each boundary. These names are FLT1, FLT2, ..., FLT6 for sides 1 through 6, respectively. If the **DEFN** subcommand is used to define other boundaries, they should be assigned names that do not conflict with those provided by the **SIDE** subcommand.
5. For INFN boundaries using the RADL option, divergence is cylindrical in 2D and spherical in 3D. For the NRML option, the pole is set by the normal distance between the reference point and the line (plane) through the element's boundary nodes.
6. RAD4 and CNVC are nonlinear boundary conditions. They may require a nonlinear thermal solution.
7. IMPD implements an impedance condition with optional spring stiffness. The density and elastic wavespeeds approximating the external medium should be entered. Ballpark estimates for the springs are K/L and G/L where K,G are bulk & shear moduli, L = cube root of model volume)

DEFN (defintion) Subcommand
BOUN (boundary) Input Group **BOUN-DEFN**

Purpose: To initiate the specification of a boundary condition for the model. The **DEFN** and **NODE** subcommands provide a general purpose alternative to the more specialized **SIDE** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **BOUN** Group: Boundary conditions are processed in the order input. The **DEFN** subcommand must precede the **DOF** and **NODE** subcommands that complete the specification of the boundary condition.

PCOM SCOM input parameters

DEFN *boundaryname type option value(1) value(2) ... value(6)*

boundaryname = (C) A unique name assigned to this boundary condition.

type = (C) Type of boundary conditions to be applied.
No default.
Any of:
- FIXD = fixed boundary
- SYMM = symmetry boundary
- ASYM = antisymmetry boundary
- ABSR = absorbing boundary
- VEL = prescribed velocity boundary
- FFLD = freefield boundary (velocity & absorber)
- ISLN = soil island boundary
- USER = user-written subroutine USRBOU is used to apply boundary conditions
- TMPR = specified temperature (thermal)
- POWR = power density (thermal)
- FLUX = prescribed heat flux (thermal)
- RADN = surface radiation condition (thermal)
- INFN = infinite element (thermal)
- RAD4 = nonlinear surface radiation condition (thermal)
- CNVC = surface convection condition (thermal)
- ASTRS = Applied Stress
- IMPD = impedance boundary (with optional spring)

option= (C) Option for the type of boundary condition specified. Valid choices for *option* depend on the choice of *type*.

option2= (C) Option for the type of boundary condition specified. Valid choices for *option* depend on the choice of *type*.

Boundary type	options
FIXD	none
SYMM and ASYM Default = ALL	TRAN- applies only to translational dofs ROT- applies only to rotational dofs ALL - applies to all dofs
ABSR and FFLD Default = NEW	STND- first order, less accurate than NEW for low CFL AUTO- Automatic optimization of boundary for solid and acoustic materials GRAZ- Grazing incidence, see note 3 below UPDT- updates absorber geometry for large deformation using STND logic UPDG- updates absorber geometry for large deformation using GRAZ option
VEL No default	FUNC - uses the time function defined with the **FUNC** command*name* - the name of a digitized time history previously specified using the **DATA HIST** option
ISLN No default	UPST - upstream boundary of soil island DNST - downstream boundary of soil island BOTM - bottom of soil island
TMPR	*option* identifies name of time history defining the temperature on the boundary nodes. Any of: FUNC - uses the time function defined with the **FUNC** command*name* - the name of a digitized time history previously specified using the **DATA HIST** option
POWR	*option* identifies name of time history defining the temperature on the boundary nodes. Any of: FUNC - uses the time function defined with the **FUNC** command*name* - the name of a digitized time history previously specified using the **DATA HIST** option
RADN	*option* identifies name of time history defining the temperature at infinity. Any of: FUNC - uses the time function defined with the **FUNC** command to define*name* - the name of a digitized time history previously specified using the **DATA HIST** option
RAD4 No default	FUNC — uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option UNIT - Constant
CNVC No default	FUNC — uses the time function defined with the **FUNC** command *name* - the name of a digitized time history previously specified via the **DATA HIST** option UNIT - Constant Option2 should be set to UNIT for a constant or the name of a digitized curve previously specified via the **DATA HIST** option.
FLUX	*option* identifies name of time history defining the temperature change between the boundary and at infinity. Any of: FUNC - uses the time function defined with the **FUNC** command*name* - the name of a digitized time history previously specified using the **DATA HIST** option
INFN	RADL — elements diverge radially NORM — elements project normally

ASTRS	*option* identifies name of time history defining the applied stress on the boundary nodes. Any of: FUNC - uses the time function defined with the **FUNC** command NAME - the name of a digitized time history previously specified using the **DATA HIST** option
USER	User-defined

value(n)= (F) The *value* inputs are applicable only to the VEL, SYMM, and ASYM boundary types.

For VEL boundaries, these values are the scale factors for each global degree of freedom, 1 through 6 (x,y,z,rx,ry,rz), respectively. The prescribed velocity for each dof of a node is the value of the prescribed time history, specified by *option*, scaled by the *value* input for that dof. The DOF subcommand can be used to control which degrees of freedom the VEL boundary applies to.

For SYMM and ASYM boundaries, typically the *value* inputs are not used. If the node or nodes for which a SYMM or ASYM boundary has been specified do not define a plane, however, then the first three *value* inputs may be used to define a vector that is normal to the desired symmetry plane.

For TMPR boundaries, *value(1)* is the scale factor for the temperature. The prescribed temperature at each node is the value of the prescribed time history, specified by *option*, scaled by *value(1).*

For POWR boundaries, *value(1)* is the scale factor for the power density. The prescribed power density in each element is the value of the time history, specified by *option*, multiplied by *value(1).*

For RADN boundaries the heat flux is given by the relation: *value(1)** T − *value(2)**T_{inf}. Typically *value(2)=value(1)*. The prescribed temperature-at-infinity is the value of the time history, specified by *option.*

For RAD4 boundaries, the heat flux is given by the relation: value(1)* (T^4 − T_{inf}^4). The prescribed temperature-at-infinity is the value of the time history, specified by *option*.

For CNVC boundaries, the heat flux is given by the relation: value(1)*h(T_{avg})*(T −T_{inf})^{expont}. The prescribed temperature-at-infinity is the value of the time history, specified by *option*. h(T_{avg}) is defined by the digitized curve specified as *option2*. Here T_{avg} = 0.5* (T + T_{inf}). The exponent is given as value(2).

For FLUX boundaries, *value(1)* is the scale factor for the heat flux. The prescribed flux at each node is the value of the prescribed time history, specified by *option*, scaled by *value(1).*

For INFN boundaries, *value(1), value(2), value(3)* are the (x,y,z) coordinates of the reference point.

For IMPD boundaries, the values are *density, p- wavespeed, s-wavespeed, (normal spring), (tangent spring.* See note 8.

Notes:
1. The **PLOD** command can be used to apply general pressure boundary conditions on the model
2. Each boundary defined with a **SIDE** subcommand is internally assigned a unique 4-character name. These names are FLX1, FLX2, ..., FLX6 for sides 1 through 6, respectively. the **DEFN** subcommand is used to define other boundaries, they should be assigned names that do not conflict with those provided by the **SIDE** subcommand. **PLOD** command can be used to apply general pressure boundary conditions on the model
3. The GRAZ option adjusts the absorber logic so that it is accurate for perfectly normal or perfectly grazing incidence. The accuracy with this option over the range of other incidence angles is poorer overall than with the default option. It should be considered only when grazing incidence waves are the primary waves impinging upon the boundary. STND is less accurate than NEW for low CFL numbers, but could be more stable in some situations.
4. For TMPR boundaries, side refers to the thermal window rather than the entire grid. As with mechanical boundaries, a unique 4-character name is assigned to each boundary. These names are FLT1, FLT2, ..., FLT6 for sides 1 through 6, respectively. If the **DEFN** subcommand is used to define other boundaries, they should be assigned names that do not conflict with those provided by the **SIDE** subcommand.
5. POWR boundaries apply incident power density throughout a volume, not flux through a surface
6. For INFN boundaries using the RADL option, divergence is cylindrical in 2D and spherical in 3D. For the NORM option, the pole is set by the normal distance between the reference point and the line (plane) through the element's boundary nodes.
7. RAD4 and CNVC are nonlinear boundary conditions. They may require a nonlinear thermal solution.
8. IMPD implements an impedance condition with optional spring stiffness. The density and elastic wavespeeds approximating the external medium should be entered. Ballpark estimates for the springs are K/L and G/L where K,G are bulk & shear moduli, L = cube root of model volume). The ABSR condition is based on the material adjacent to the boundary. This condition can be used to represent the same material or a different material.

DOF (degrees of freedom) Subcommand
BOUN (boundary) Input Group **BOUN-DOF**

Purpose: To select a subset of the possible global degrees of freedom for nodes to which this boundary applies. This command is valid only for FIXD and VEL boundary conditions.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **BOUN** Group: Must follow the **DEFN** subcommand and precede the **NODE** subcommand for the boundary with which it is associated.

PCOM	SCOM	input parameters
	DOF	*dof(1) dof(2) ... dof(6)*

dof(n) = (C) Global degrees of freedom to which the current boundary specification applies. Any of: X, Y, Z, RX, RY, RZ that are the three global translation and rotation degrees of freedom of the nodes. Up to 6 dofs may be input.

Note: Specification of degrees of freedom of the model that are inactive, i.e., the z-component for 2D plane strain models is ignored.

NODE Subcommand
BOUN (boundary) Input Group **BOUN-NODE**

Purpose: To define the range of nodal indices to which the boundary condition specified by the previous **DEFN** subcommand are applied. More than one **NODE** subcommand can follow a single **DEFN** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **BOUN** Group: Must follow the **DEFN** subcommand with which it is to be associated.

PCOM	SCOM	input parameters
	NODE	*ibegin iend jbegin jend kbegin kend (matname)*

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the boundary region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the boundary region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the boundary region.
Default: *kbegin* = 1, *kend* = maximum K

matname = (C) For TMPR boundaries, only apply BC if and adjacent element is of type *matname*

Notes: 1. For *boundarytype* of SYMM, ASYM, ABSR, and ISLN, the **NODE** subcommand must define a plane of nodes. Therefore, *ibegin* = *iend*, *jbegin* = *jend*, or *kbegin* = *kend*.
2. For *boundarytype* of VEL or FIXD, the **NODE** subcommand may define any region of nodes.
3. For SYMM and ASYM boundary conditions all nodes specified should be coplanar. If the normal to the symmetry plane was default by the **DEFN** subcommand, then it is computed using the four corners (two ends for 2D models) of the defined surface. If the four corners do not define a plane, then the user must have specified the surface normal on the **DEFN** subcommand.

MSHP (mode shape) Subcommand
BOUN (boundary) Input Group **BOUN-MSHP**

Purpose: To specify a mode shape for the **POWR** boundary condition. This provides an alternative to the **NODE** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **BOUN** Group: Boundary conditions are processed in the order input. The **DEFN** subcommand must precede the **MSHP** subcommands that complete the boundary condition specification.

PCOM	SCOM	input parameters
	MSHP	*pathname arrayname nquad*
or		
	MSHP	MATR *matname*
or		
	MSHP	*pathname asxx asyy aszz asxy asyz asxz nquad*

pathname = (C) The complete pathname for the *flxdato* file containing the modeshape. No default.

arrayname= (C) The name of the modeshape array on the *flxdato* file. No default.

arrayname= (C) The names of the applied stress arrays on the *flxdato* file. *asyz* and *asxz* should be defaulted for 2D No default.

nquad= (I) The number of quadrature points in each direction. Default = 1.

matname= (C) Material name to receive applied power. Mode shape is 1.0 for this material, zero for all others.

Notes:
1. The flxdato file must contain coordinate information (ie, **DATA OUT** MODL must be specified in creating the modeshape file.
2. The modeshape must be standard partition element data, but the discretization does not need to match that of the current model.
3. If the thermal model elements are large compared to the mechanical model, it may be adviseable to use multiple quadrature points per element in order to accurately calcuate the power deposition in each thermal element.
4. The power deposition is stored in an array named *boundaryname* where *boundryname* is taken from the DEFN command.
5. The Applied stresses must be standard partition element data, but the discretization does not need to match that of the current model.
6. If the model elements are large compared to the applied stress data, it may be advisable to use multiple quadrature points per element in order to accurately calculate the power deposition in each element.

PLOT Subcommand
BOUN (boundary) Input Group **BOUN-PLOT**

Purpose: To control the generation of "printer plots" of applied boundary condition time histories. By default, the print file for the job contains a simple printer plot for each prescribed time boundary time history. These plots are generated at the end of the **BOUN** command group. This command allows the user to suppress these plots.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **BOUN** Group: None.

PCOM SCOM input parameters

PLOT *option*

option = (C) Option to control the generation of printer plots for user-prescribed boundary condition time histories. Set to either YES or NO. If set to NO, no printer plots are generated. If set to YES, printer plots are generated.
Default = YES.

CALC (calculate) Input Command

CALC

Purpose: To specify secondary field variables and energy quantities to be computed and stored during execution. Secondary field variables are nodal and elemental quantities that the computational algorithm does not require to be stored during the computation.

Use Is: Optional.

Multiple Use: Usually not allowed.

Order Dependence: Must precede **PRCS** command, with some exceptions: TFRM, MAX, LOSS.

PCOM	SCOM	input parameters
CALC		
	AINTN	
	ARST	
	AVRG	*avrgname arrayname weightoption region ibeg iend jbeg jend kbeg kend*
	BWORK	*surfacename ibeg iend jbeg jend kbeg kend normal*
	CURL	
	DISP	*component component component*
	ECENT	*elmarray nodarray (RATE irate)*
	ENRG	*energyname (option) ibegin iend jbegin jend kbegin kend*
	ENRGINTG	*(rate) name etype rtype ibeg iend jbeg jend kbeg kend*
	EQEP	*options*
	ESTP	*option*
	FRAG	*fragname ibegin iend jbegin jend kbegin kend bodyfile nadloop*
	INTEXT	*dataname type filename ncolbegin ncolend nskip*
	INTG	*dataname intgname*
	LOSS	*option*
	MATPROP	*arrayname matprop*
	MAX	*dataname minname maxname*
	MAXE	
	P2IN	
	PRES	
	PTEN	*elmtype (ilayer) tensor (compres) name (RATE irate) ibegin iend jbegin jend kbegin kend*
	PTEN	HEX *tensor (compres) name (vector) (RATE irate) nhexbeg nhexend*
	PTEN	TET *tensor (compres) name (vector) (RATE irate) ntetbeg ntetend*
	PZENRG	*ioptk ioptd iopte ioptc*
	RESULTANT	XYZ *x0 y0 z0 v11 v21 v31 v12 v22 v32 side*
	RESULTANT	IJK *i1 j1 k1 i2 j2 k2 i3 j3 k3 side*
	RESULTANT	GCON *n1 n2 n3 side*

PCOM SCOM	input parameters

CALC (cont.)

ROTN	
SJ2P	
STRN	*component component component component component component*
STRS	
TIMER	*option*
TFRM	*option (datanames)* TO *axisname ibegin iend jbegin jend kbegin* & *kend*
VELM	
VMAG	*vmagname xdataname ydataname (zdataname)*
VOLM	*volumename (histname) ibegin iend jbegin jend kbegin kend*
VOLMINTG	*(rate) name array ibeg iend jbeg jend kbeg kend*
VOLR	
VPHASE	*datanamefrom datanameto (fmin fmax nf vmin vmax nv)*
END	

Notes: 1. If snapshots or time histories of secondary field variables are desired for any stations in the grid, these variables must be specified with a **CALC** command. They are computed and stored in memory for all stations in the grid.

2. Computing secondary field variables requires relatively little cpu time, but large amounts of memory may be required to store the requested data.

AINTN (Acoustic Intensity) Subcommand
CALC (Calculate) Input Group **CALC-AINTN**

Purpose: To specify that the acoustic intensity be computed.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
Within **CALC** Group: None.
Command must be entered after element definitions and before execution.

PCOM	SCOM	input parameters
	AINTN	

Notes: 1. Computes the instantaneous acoustic intensity (pressure * velocity) and stores vector components in arrays XAINT, YAINT and for 3D: ZAINT. The vector magnitude is stored in MAINT.

ARST (acoustic radiation stress) Subcommand
CALC (Calculate) Input Group **CALC-ARST**

Purpose: To specify that an acoustic radiation stress tensor be computed.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
Within **CALC** Group: None.
Command must be entered after element definitions and before execution.

PCOM	SCOM	input parameters
	ARST	

Notes: 1. Computes an estimate of the time-averaged "Acoustic Radiation Stress" in the beam direction due to an oscillating signal. This is a very small DC value that arises due to the nonlinearity of the pressure vs. density relation. Because it is DC, it can produce motion over a period of time. See for example, "Ultrasound in medicine," Duck, F., A. Baker and H. Starritt, eds, Institute of Physics Publishing, Bristol, UK, 1999.

2. Procedure is to bring problem to steady state, CALC ARST, then execute a number of cycles. The time average will converge to a constant for an exact number of cycles, or as the number increases. The tensor values are in ARSXX, ARSYY, ARSZZ, ARSXY, and for 3D: ARSYZ and ARSXZ.
3. This option can be used in conjunction with the BOUN ASTRS command to apply radiation forces. DATA OUT the ARSXX, etc. arrays. These can then be modulated in time (e.g. with a duty cycle) and used to drive a second model.

AVRG (average) subcommand
CALC (calculate) input group **CALC-AVRG**

Purpose: To request that the time-varying sum and average of nodal or elemental information be computed during a calculation.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM SCOM input parameters

AVRG *avgname arrayname weightoption region ibeg iend jbeg jend kbeg kend*
or
avgname arrayname weightoption NOD2 mat1 mat2 rootopt ibeg iend jbeg & jend kbeg kend

avgname = (C) Name of this average group. If *avgname* is the same as that defined on a preceding **AVG** subcommand, then this command will append the specified region to those previously assigned to *avgname*. No default.

arrayname = (C) Array containing elemental or nodal information to be averaged. No Default.

weightoption = (C) Weighting to use for computing average. Any of UNIT or MASS for nodal arrays; UNIT or VOLUME for elemental arrays. Default = UNIT.

region = (C) Spatial region over which to compute average. Enter REGN for an ijk region or NOD2 for an interface between materials.

ibegin..., kend = (I) Beginning and ending ijk-indices of nodes which bound the region for average computation.
Default: *ibegin* = 1, *kend* = maximum K

mat1, mat2 = (C) Materials bounding an interface. Nodal arrays will be collected on the interface, elemental arrays will be on the mat1 side

rootopt = *(C)* Enter * for mat1 and mat2 only. Root1 for all materials with names beginning with mat1, or root2 for all materials with names beginning with mat2

Notes: 1. The AVRG(3,n) data array is created when this option is used. AVRG(1,i) contains the weighted average, AVRG(2,i) the weighted sum, and AVRG(3,i) the sum of the weights for average i.
2. AVGN(4,n) is also created. AVGN(1,i) is user-assigned name of the average. AVGN(2,i) is the array being averaged.

BWORK (boundary work) Subcommand
CALC (calculate) Input Group **CALC-BWORK**

Purpose: Calculate work and instantaneous power flux through specified surfaces.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

BWORK *surfacename ibeg iend jbeg jend kbeg kend normal*

surfacename = (C) Unique name for this surface. May be repeated to integrate over multiple subsurfaces.
No default.

ibeg ,iend = (I) Nodal i range of subsurface.
No default.

jbeg, jend = (I) Nodal j range of subsurface.
No default.

kbeg, jend = (I) Nodal k range of subsurface.
No default.

normal = (I) Outer normal. Enter 1 or -1. The outer normal for the subsurface is implicitly defined as the positive i, j or k direction not in the surface. Enter -1 to reverse the sign. Default = 1.

Notes: 1. Each subsurface should be a surface in 3D or a line in 2D. So *ibeg=iend* or *jbeg=jend* or *kbeg=kend* in 3D. Likewise *ibeg=iend* or *jbeg=jend* in 2D.
2. The array BWORKN(i) contains the name of each surface. BPOWR(i) has the instantaneous power flux through this surface and BWORK(i) contains the time integral of BPOWR(i).

CURL Subcommand
CALC (calculate) Input Group **CALC-CURL**

Purpose: To request that the curl of the displacement field and the volumetric strain be computed for each continuum element during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **CALC** Group: None.

PCOM	SCOM	input parameters
	CURL	-none-

Note: The components of the curl vector are stored in the XCRL(i,j,k), YCRL(i,j,k), and ZCRL(i,j,k) data arrays. (Note that only the ZCRL array is present for 2D axisymmetric and plane strain models). The volumetric strain values are stored in the EVOL(i,j,k) data array. The ijk indices for these arrays correspond to elemental indices.

DISP (displacement) Subcommand
CALC (calculate) Input Group **CALC-DISP**

Purpose: To specify which displacement components are to be computed during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	DISP	*component component component*

component = (C) Component of displacement to compute.
May be any of: X, Y, or Z.
If all *component* inputs are left blank, the default is to compute all active displacement components.
Naming of an inactive displacement component is ignored.

Note: X, Y, and Z displacement data values are stored in the XDSP(i,j,k), YDSP(i,j,k), and ZDSP(i,j,k) data arrays, respectively. The ijk indices of these arrays correspond to nodal indices.

ECENT (element-centered) Subcommand
CALC (calculate) Input Group **CALC-ECENT**

Purpose: To interpolate nodal array values to element centers.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

ECENT *elmarray nodarray name (RATE irate)*

elmarray = (C) Name of element array to be created.
No default.

nodarray = (C) Name of existing nodal array to interpolate element-centered values from.
No default.

irate = (I) Optional input. By default, the computation will be executed only once. If the RATE *irate* input is entered, the values will be computed every *irate* timestep

ENRG (energy) Subcommand
CALC (calculate) Input Group **CALC-ENRG**

Purpose: To compute the translational momentum and kinetic energy for a group of nodes.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM SCOM input parameters

ENRG *energyname* (*option*) *ibegin iend jbegin jend kbegin kend*

energyname = (C) Name of the energy region. Specifying the same *energyname* more than once on consecutive **ENRG** subcommands will append the nodal region to those specified previously. No default.

option = (C) Optional input parameter. If specified, defines the way in which the energy will be calculated, see note below. May be set to any of: SPLT or TOTL. If SPLT, energy is split into contributions due to +velocity nodes and those due to - velocity nodes. The TOTL option produces totals without regard to the sign of a nodes velocity. If not input, TOTL is assumed.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the region. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the region. Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the region. Default: *kbegin* = 1, *kend* = maximum K

CALC-ENRG

Note: This command computes the translational momentum and kinetic energy of one or more nodal regions and keeps track of the peak values during an analysis. The information is stored in the data manager array ENRD(i,j) depending on *option*. The i-index of this array ranges from 1 to 9.The j-index ranges from 1 to the number of energy regions input.

	option=TOTL	*option*=SPLT
i = 1	current x-momentum	current +x-momentum
i = 2	current y-momentum	current +y-momentum
i = 3	current z-momentum	current +z-momentum
i = 4	current kinetic energy	current + kinetic energy
i = 5	peak x-momentum to this point in time	current -x-momentum
i = 6	peak y-momentum to this point in time	current -y-momentum
i = 7	peak z-momentum to this point in time	current -z-momentum
i = 8	peak kinetic energy to this point in time	current - kinetic energy
i = 9	total translational mass for this energy set	(same as *option*=TOTL)

ENRGINTG (Energy Integral) Subcommand
CALC (Calculate) Input Group **CALC-ENRGINTG**

Purpose: To calculate internal energy per unit volume and optionally, it's volume integral over a specified window.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **CALC** Group: None.

PCOM SCOM input parameters

ENRGINTG *(rate) name etype rtype ibeg iend jbeg jend kbeg kend*

rate = (I) Optional. If entered, the integral is calculated every *rate* timesteps. If *rate* is not entered, the integral is calculated only at the current timestep.

name = (C) Unique name identifying the energy integral.

etype = (C) Element types to be considered. Any of: ALL, CONT, BEAM, SHEL, BAR, FUSE, BOLT.

vtype = (C) Volume type. Enter REGN, or NONE. Default = REGN. See Note 1.

ibegin, iend = (I) Beginning and ending I-indices of nodes which bound the interval.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes which bound the interval.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes which bound the interval.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. Multiple ENRGINTG subcommands with the same *name* can be entered to associate multiple ijk windows with the same *name*. The resulting integral is the sum of the integrals over each ijk range. The values of *rate* and *array* are taken from the last entry of a given *name*.
2. Integral values are stored in data manager array ENRINT(*n*) in the order input where *n* is the number of *names*. *name* is stored in order in data manager array ENRNAME(*n*).
3. If NONE is entered, the Internal energy per unit volume is computed for each element, but no integral is computed
4. Internal Energy per unit volume is stored in the arrays energy_ct, energy_sh, energy_ba, energy_b3 and energy_ln for continuum, shell, bar, beam and bolt elements respectively.
5. This option is not yet implemented for GCON Hex elements.

EQEP (equivalent strain) Subcommand
CALC (calculate) Input Group **CALC-EQEP**

Purpose: To specify that element equivalent strain values be computed during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

EQEP *options*

options = (C) One or more of: TOTAL, PLASTIC
TOTAL = compute equivalent total strain
PLASTIC = compute equivalent plastic strain
No default.

Note: The equivalent total strain values are stored in the EQTE(i,j,k) data group and the equivalent plastic strain values are stored in the EQPE(i,j,k) data group. The ijk indices for these arrays correspond to elemental indices.

ESTP (element timestep) Subcommand
CALC (Calculate) Input Group **CALC-ESTP**

Purpose: Calculate and save an array containing the maximum stable timestep for each continuum element.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **CALC** Group: None.

PCOM SCOM input parameters

ESTP *option*

option = (C) Enter ACTUAL to compute timestep, SQUARED for timestep squared. Default = SQUARED

Notes:
1. The ESTP array is always created and updated when Large Deformation is active. By default, it is not computed in small deformation.
2. By default ESTP contains dt^2, where dt is the maximum stable timestep. This is computationally more efficient. Flex will limit the timestep to *stbfac*dt* where *stbfac* is specified on the TIME command.

FRAG (fragments) Subcommand
CALC (calculate) Input Group **CALC-FRAG**

Purpose: To inpsect the model for fragments: nodes that have mass but are not connected to any active elements. Information about these fragments is printed and the locations are stored for plotting with the **GRPH PLOT** FRAG command. Isolated bodies are optionally output to a file.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
FRAG		*fragname ibegin iend jbegin jend kbegin kend (bodyfile) (nadloop)*
	or	
FRAG		CLEAR

fragname = (C) Name of this fragment group. If not default, this command appends the new fragment group information to previously computed fragment groups.
If set to CLEAR, then clears out all previous fragment groups and any following ijk-indices are ignored.
Default = clears all previous fragment groups and calculates new fragment group named FRAGMENT.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the region to be checked for fragments.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the region to be checked for fragments.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the region to be checked for fragments.
Default: *kbegin* = 1, *kend* = maximum K

bodyfile = (C) Name of file to output information on all bodies in the model at the time the **CALC-FRAG** subcommand is issued. Default is no body file generated. See notes 6 and 7.

nadloop = (I) Number of additional loops for convergence of number of bodies calculated. Default = 0

CALC-FRAG

Notes: 1. The **FRAG** subcommand reports the number of fragment nodes and their total mass in the print file. Once this command is input, the plottable data array FRAG are present in the code. The data can be plotted with the command
GRPH PLOT FRAG. Each fragment is shown by drawing a box centered at its location.

2. If different fragment names are used for different ijk regions of the mesh, the fragments are plotted using different colors for each fragment group.
3. The size of the box drawn to represent each fragment is controlled by the **GRPH PSET** BOXSIZE option. Additionally, the aspect ratio of the box can be controlled by the **GRPH PSET** BOXSCALE option.
4. Data for each set of fragement nodes are contained in the FRAGD(i,j) data array. The j-index ranges from 1 to the number of fragment sets. The i-index ranges from 1 to 7:
   i=1 > number of fragment nodes for this set
   i=2 > total mass attached to the fragment nodes for this set
   i=3 > total x-momemtum of these fragment nodes
   i=4 > total y-momemtum of these fragment nodes
   i=5 > total z-momemtum of these fragment nodes
   i=6 > lowest velocity magnitude for any node in this fragment set
   i=7 > highest velocity magnitude for any node in this fragment set
5. The code checks for beam,bar,shell, and continuum element connectivity in deciding if a node is a fragment. Information in *bodyfile* consists of the number of separate bodies (single nodes or node groups connected by continuums, shells, beams, and/or bars), as well as the mass, the x,y,z location and the velocity of each body.
6. If *bodyfile* already exists, it is overwritten with new data. This option is implemented only for skewed meshes.
7. To identify individual bodies, element connectivity is determined by propagating the lowest node id in a body to other connected nodes. Two forward loops and one backward loop are executed automatically, resulting in a fairly accurate estimate of the bodies in the model. A more accurate list for models containing long, thin bodies may be obtained by specifying additional loops with *nadloop*.

INTEXT Subcommand
DATA Input Group**DATA-INTEXT**

Purpose: To assign data values which are stored on an external file to an internal Data Manager array. The array may or may not already exist. The format of the external file is rows and columnsof text.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM input parameters

INTEXT *dataname type filename ncolbegin ncolend nskip*

dataname = (C) The data manager array name that will contain the data once it has been imported from the Data Input File. No default.

type= (C) Any of: C for character data, I for integer data or F for floating point data. No default.

filename = (C) name of file to be read. No default.

ncolbegin, ncolend= (I) Beginning and ending columns of data to be read into dataname. Default is all columns.

nskip = (I) Number of lines to skip before reading data. Default = 0. See note 1.

Notes: 1. For Integer or Real types, column labels are automatically skipped.

INTG (integral option) Subcommand
CALC (calculate) Input Group **CALC-INTG**

Purpose: To request that the time integral of real valued data contained in a data manager array be computed during the analysis. The result of this option is to produce an array having the same size as the original but whose contents contain the time integral of each array member.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

INTG *dataname intgname*

dataname= (C) Name of the data array whose time integral will be computed. Include path if not in top directory.
No default.

Intgname= (C) User-defined name of the data array that will contain the integrated data values of *dataname*.
Default = NONE, integrated data are not computed.

Notes:
1. Any real number data group may be accessed with this option.
2. The *intgname* array is the same size as the *dataname* array, consequently significant memory resources may be required to store the data created with this option.
3. As an example, for a 3D array, the value stored in *intgname*(1,1,1) would be the time integral of *dataname*(1,1,1) integrated up to the present simulation time.
4. *intgname* will placed in the same directory as *dataname*. The path should not be included for *intgname*.

LOSS (energy loss) Subcommand
CALC (calculate) Input Group **CALC-LOSS**

Purpose: To compute accumulated energy dissipation per unit volume. Currently implemented only for VDMP, SDMP, MDMP, RDMP and NWTN damping models. Implemented for small deformation 2D axisymmetric, 2D plane strain and 3D processors. Command can be issued before the **PRCS** step or at any time during the calculation.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

LOSS *option*

option = (C) Any of: ON, OFF
ON = compute energy loss during following time execution commands
OFF = do not compute energy loss during following time execution commands
Default = ON.

Notes:
1. This command integrates the power dissipation per unit volume from the time an ON command is issued until an OFF command is encountered.
2. When an ON command is encountered, the accumulated value is reset to zero.
3. The dissipated energy per unit volume values are stored in the LOSS (i,j,k) array. Where the ijk-indices are elemental indicies.

MATPROP (Material Properites) Subcommand
CALC (Calculate) Input Group **CALC-MATPROP**

Purpose: To load an array with the value of a selected material property — primarily for visualization.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **CALC** Group: None.

PCOM SCOM input parameters

MATPROP *arrayname matprop*

arrayname = (C) Name of the array to be created. Will be dimensioned to the number of elements in the model.
No default.

matprop = (C) The material property to be loaded. See note 1.
No default.

Notes: 1. The available material property names are listed in the flxprt file after the MATR input section. E.g. CALC MATPROP *shear shr*. Will create an array *shear* containing the shear modulus of each element.

MAX (maximum option) Subcommand
CALC (calculate) Input Group **CALC-MAX**

Purpose: To specify that maximum and/or minimum value arrays be computed for each member of a data group throughout the execution phase of a problem. This option provides a simple way to obtain distributions of peak values of stress, velocity, etc. over the duration of a calculation for the entire model. See also the **MAXE** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

MAX *dataname minname maxname*

dataname= (C) Name of the data group for which minimum or maximum data values are to be evaluated. Include path if not in top directory. No default.

minname= (C) User-defined name of the data group containing the minimum data values of *dataname.* Set to NONE if minimum data values are not desired.
Default = NONE, no minimum data array is computed.

maxname= (C) User-defined name of the data group containing the maximum data values of *dataname.* Set to NONE if maximum data values are not desired.
Default = NONE, no maximum data array is computed.

Notes:
1. Any real number data group may be accessed with this option.
2. The *minname* and *maxname* arrays are the same size as the *dataname* array; consequently, significant memory resources may be required to store the data created with this option.
3. The values stored in *minname* are actually the smallest negative values that occur over the duration of the calculation, and the values stored in *maxname* are the largest positive values that occur.
4. *minname* and *maxname* are placed in the same directory as *dataname.* The path should not be included for *minname* and *maxname.*

MAXE (maximum strain) Subcommand
CALC (calculate) Input Group **CALC-MAXE**

Purpose: To specify that minimum and maximum strain values be determined for each element in the model. This option provides the same functionality as the **MAX** subcommand automatically for all strain components. If only certain components are of interest, the **MAX** subcommand may be used.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	MAXE	-none-

Notes: 1. Maximum and minimum strain values are determined only for the active strain components.
2. The maximum and minimum strain values are stored in the following data arrays whose ijk indices correspond to elemental indices:

Strain Component	Minimum Values	Maximum Values
ε_{xx}	MNXX(i,j,k)	MXXX(i,j,k)
ε_{yy}	MNYY(i,j,k)	MXYY(i,j,k)
ε_{zz}	MNZZ(i,j,k)	MXZZ(i,j,k)
ε_{xy}	MNXY(i,j,k)	MXXY(i,j,k)
ε_{yz}	MNYZ(i,j,k)	MXYZ(i,j,k)
ε_{xz}	MNXZ(i,j,k)	MXXZ(i,j,k)

P2IN (pressure) Subcommand
CALC (calculate) Input Group **CALC-PRES**

Purpose: To specify that the time integral of pressure values be computed during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	P2IN	*-none-*

Notes:
1. The P2IN data values are stored in the P2IN(i,j,k) data group. The ijk indices for this array corresponds to elemental indices.
2. This option also activates the calculation of PRES. P2IN is useful in approximating acoustic radiation pressure.

PRES (pressure) Subcommand
CALC (calculate) Input Group **CALC-PRES**

Purpose: To specify that element pressure values be computed during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

PRES *option*

option = (C) Enter acoustic to open an additional APRS array containing -pres .
Default: aprs not calculated.

Notes:
1. The pressure data values are stored in the PRES(i,j,k) data group. The ijk indices for this array corresponds to elemental indices.
2. The computed value of pressure, p, is the hydrostatic stress of an element:
$$p = \frac{1}{3}[\sigma_{xx} + \sigma_{yy} + \sigma_{zz}]$$
3. PRES is negative in compression. APRS is positive in compression – more common in the ultrasound community.

PTEN (principal tensile) Subcommand
CALC (calculate) Input Group **CALC-PTEN**

Purpose: To compute the principal tensile values of stress or strain for a group of elements throughout a computation and store them in a data array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM SCOM input parameters

PTEN *elmtype (ilayer) tensor (compres) name (RATE irate) &*
ibegin iend jbegin jend kbegin kend
or
PTEN HEX *tensor (compres) name (vector) (RATE irate) nhexbeg nhexend*
or
PTEN TET *tensor (compres) name (vector) (RATE irate) ntetbeg ntetend*

elmtype = (C) Any of: CONT, SHEL, FUSE, HEX, TET
CONT = compute values for ijk continuum elements
SHEL = compute values for shell elements (see note 1)
FUSE = compute values for FUSE elements
HEX =compute values for general connectivity hex elements
TET =compute values for general connectivity tet elements
No default.

ilayer = (I) Optional input. If *elmtype* = SHEL, then *ilayer* may be input to define a specific shell layer number to compute the values for. If not input, the maximum value for all layers of the shell will be computed.

tensor = (C) Any of: STRS, STRN
STRS = compute values for stress tensor
STRN = compute values for strain tensor
No default.

compres = (C) Optional input. If COMP is entered, minimum (compressive) values will be computed rather than maximum (tensile) values.

name = (C) Name of data array which will contain the computed values. No default.

(continued)

vector = (C) Optional input. If VECT is entered, the eigenvector corresponding to the requested principal value is computed. See note 2 and note 3.

irate = (I) Optional input. By default, the computation will be executed only once. If the RATE *irate* input is entered, the values will be computed every *irate* timesteps.

ibegin, iend = (I) Beginning and ending I-indices of nodes which bound the region of elements to be computed. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes which bound the region the region of elements to be computed. Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes which bound the region of elements to be computed. Default: *kbegin* = 1, *kend* = maximum K

nhexbeg, nhexend = (I) Beginning and ending global general connectivity hex element numbers of range of elements to be computed. Default: *nhexbeg* = 1, *nhexend* = maximum hex element number

ntetbeg, ntetend = (I) Beginning and ending global general connectivity tet element numbers of range of elements to be computed. Default: *ntetbeg* = 1, *ntetend* = maximum tet element number

---

Notes: 1. For shell strains, in-plane and transverse shear components are considered.

2. Due to the computational expense of computing the eigenvectors, eigenvectors can only be computed at the time the command is executed and will not be computed if the *irate* parameter is entered.

3. Eigenvector components are stored in three arrays of size *n*, where *n* is the number of elements of type *elemtype*. The vectors are named *name*:x, *name*:y and *name*:z for the three components of each eigenvector. If a sub-region of elements is specified, only vectors within the sub-region are computed. Eigenvectors can be plotted using the GRPH ARROW SHEL command for shell elements and the GRPH ARROW CSTM command for continuum elements.

PZENRG (piezoelectric energy) Subcommand
CALC (calculate) Input Group **CALC-PZENRG**

Purpose: To calculate the various energies for piezoelectric models.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None, but must follow **PIEZ** command.

PCOM	SCOM	input parameters
	PZENRG	*ioptk ioptd iopte ioptc*

ioptk = (C) Flag to calculate kinetic energy per unit volume. Either ON or OFF. Default = ON

ioptd = (C) Flag to calculate dielectric energy. Either ON or OFF. Default = ON

iopte = (C) Flag to calculate elastic (potential) energy. Either ON or OFF. Default = ON

ioptc = (C) Flag to calculate piezoelectric coupling energy. Either ON or OFF. Default = ON

Notes: 1. All energy calculations are performed at the element center at the half-timestep.

2. Kinetic energy per unit volume, dielectric energy, elastic energy and coupling energy are calculated as

$$E_K = \frac{1}{2}\rho \mathbf{v}\cdot\mathbf{v}\ , \quad E_D = \frac{1}{2}\mathbf{E}\cdot\boldsymbol{\varepsilon}\cdot\mathbf{E}\ , \quad E_E = \frac{1}{2}\mathbf{S}:\mathbf{C}^{\mathbf{E}}:\mathbf{S}\ , \quad E_C = \frac{1}{2}\mathbf{E}\cdot\mathbf{e}:\mathbf{S}$$

where $\mathbf{v}$ are the velocities, ρ is the mass density, $\mathbf{E}$ the electric fields, $\mathbf{S}$ the strains, $\mathbf{e}$ the coupling coefficients, $\boldsymbol{\varepsilon}$ the dielectric tensor and $\mathbf{C}^{\mathbf{E}}$ the Elastic stiffness at zero electric field.

3. Kinetic energy is stored in data manager array CLPK(i,j,k), dielectric energy is stored in CLPD(i,j,k), elastic energy is stored in CLPE(i,j,k) and coupling energy is stored in CLPP(i,j,k). The ijk indices correspond to elemental indices for these arrays.

RESULTANT Subcommand
CALC (Calculate) Input Group **CALC-RESULTANT**

Purpose: To calculate resultant forces and moments on a cutting plane.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: None, can be entered before or after PRCS step.
Within **CALC** Group: None.

PCOM SCOM input parameters

RESULTANT XYZ *x0 y0 z0 v11 v21 v31 v12 v22 v32 side*
or
RESULTANT IJK *i1 j1 k1 i2 j2 k2 i3 j3 k3 side*
or
RESULTANT GCON *n1 n2 n3 side*

For all options: Inputs define a base point and two vectors to define the cutting plane. The length of each vector defines the length of a side of the cutting plane. The cutting plane will be a parallelogram, but *v1* and *v2* are not required to be orthogonal. The local 3-direction (*v3*) is defined by $v1 \times v2$. The local 2-direction is defined by $v3 \times v1$.

sideopt = (C) Option to define what side of cutting plane to get element stresses from if the cutting plane intersects a node. Either of:
POS = shift cutting plane ε in direction of positive normal of the plane
NEG = shift cutting plane ε in direction of negative normal of the plane
No default.

For XYZ option: Define a base point and two vectors to define the cutting plane. The cutting plane is stationary in time.

x0, y0, z0 = (F) Coordinates of the base point of a stationary cutting plane. See Diagram. No default.

v11, v21, v31 = (F) Components of vector *v1* of a stationary cutting plane. See diagram. No default.

v12, v22, v32 = (F) Components of vector *v2* of a stationary cutting plane. See diagram. No default.

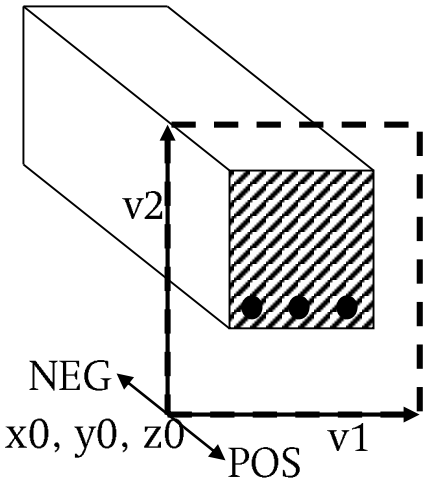
(description of **RESULTANT** subcommand continues on next page)

For IJK option: Define three points to define the cutting plane. Point 1 is the base point. The local 1-direction (*v1*) is defined by the vector from point 1 to point 2. A vector *v2* is defined from point 1 to point 3. The cutting plane location and orientation is updated each time step based on the new locations of the nodes defining it.

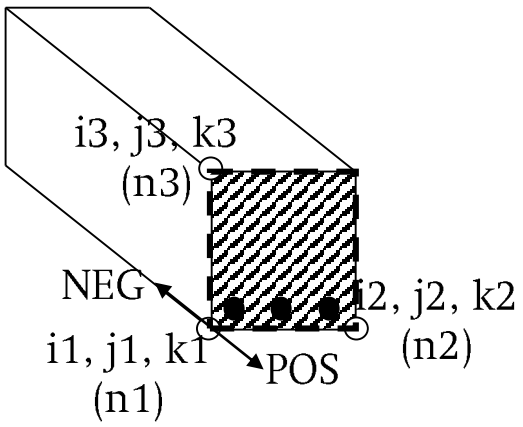
i1, j1, k1, ..., i3, j3, k3 = (F) IJK indices of three points defining a moveable cutting plane. See Diagram. No default.

For GCON option: Define three points to define the cutting plane. Point 1 is the base point. The local 1-direction (*v1*) is defined by the vector from point 1 to point 2. A vector *v2* is defined from point 1 to point 3. The cutting plane location and orientation is updated each time step based on the new locations of the nodes defining it.

n1, n2, n3 = (F) General connectivity node numbers of three points defining a moveable cutting plane. See Diagram. No default.

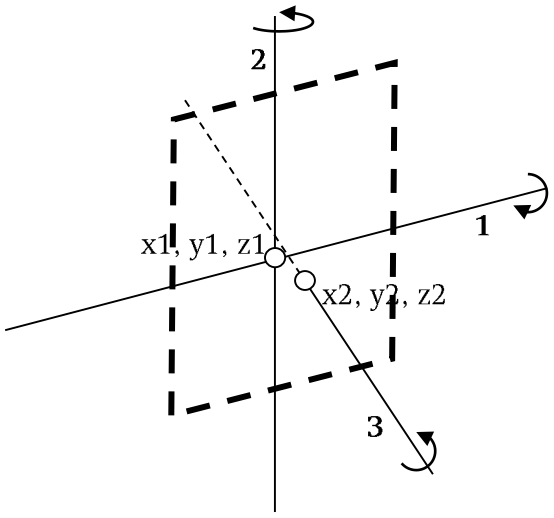


Stationary cutting plane: option XYZ



Moveable cutting plane: options IJK and GCON

(description of **RESULTANT** subcommand continues on next page)



Neutral points for in-plane moments (x1, y1, z1) and moment about plane normal (x2, y2, z2)

Notes: 1. For XYZ option the cutting plane is stationary in time. For IJK and GCON options the cutting plane is attached to the structure and moves over time with the three nodes that define it.
2. Resultant forces on the cutting plane are stored in array RESFRC(i,j), where i = 1, 2, 3 in the cutting plane local system and j = 1 to number of resultant cutting planes.
3. Resultant moments on the cutting plane about the neutral axis are stored in array RESMOM(i,j), where i = 1, 2, 3 in the cutting plane local system and j = 1 to number of resultant cutting planes.
4. Global coordinates of the neutral points (points of intersection of the neutral axes on the cutting plane) for the in-plane moments and out-of-plane moment are stored in RESNPT(i,j), where i = x1, y1, z1, x2, y2, z2 and j = 1 to number of resultant cutting planes. See diagram. The neutral point coordinates are updated at each time step based on the force distribution on the cutting plane, but assumes that the cutting plane remains plane and undeformed.
5. Currently, all IJK continuum, general connectivity hexahedral, shell, membrane, beam and bar elements that intersect a cutting plane are considered when determining the resultant forces and moments.

ROTN (rotation) Subcommand
CALC (calculate) Input Group

Purpose: To specify which (if any) nodal rotation components are to be computed during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	ROTN	*component component component*

component = (C) Component of displacement to compute.
May be any of: X, Y, or Z.
If all *component* inputs are left blank, the default is to compute all active rotation components.
Naming of an inactive rotation component is ignored.

Note: X, Y, and Z rotation data values are stored in the RXDP(i,j,k), RYDP(i,j,k), and RZDP(i,j,k) data arrays, respectively. The ijk indices of these arrays correspond to nodal indices.

SJ2P ($\sqrt{J'_2}$)Subcommand
CALC (calculate) Input Group **CALC-SJ2P**

Purpose: To specify that the square root of the second invariant of the deviatoric stress tensor be computed during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	SJ2P	-none-

Note: The $\sqrt{J'_2}$ data values are stored in the SJ2P(i,j,k) data group. The ijk indices for this array corresponds to elemental indices.

STRN (strain) Subcommand
CALC (calculate) Input Group **CALC-STRN**

Purpose: To specify which strain components are to be computed during the analysis. Strains values are tensor strain quantities. For large deformation runs, strains are accumulated infinitesimal strain increments with rigid body rotations accounted for.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **CALC** Group: None.

PCOM	SCOM	input parameters
	STRN	*component component component component component component*

component = (C) Component of strain to compute.
May be any of: XX, YY, ZZ, XY, YZ, or XZ.
If all *component* inputs are left blank, the default is to compute all active strain components.
Naming of an inactive strain component is ignored.

Note: For large deformation calculations, if any strain components are reqested, all strain components are computed.

Note: XX, YY, ZZ, XY, YZ, and XZ strain data values are stored in the EPXX(i,j,k), EPYY(i,j,k), EPZZ(i,j,k), EPXY(i,j,k), EPYZ(i,j,k), and EPXZ(i,j,k) data arrays, respectively. The ijk indices of these arrays correspond to elemental indices.

STRS (stress) Subcommand
CALC (calculate) Input Group **CALC-STRS**

Purpose: To request that stress values be computed for models that do not automatically require the stress arrays during the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	STRS	-none-

Notes: 1. XX, YY, ZZ, XY, YZ and XZ stress data values are stored in theSGXX(i,j,k), SGYY(i,j,k), SGZZ(i,j,k), SGXY(i,j,k), SGYZ(i,j,k), and SGXZ(i,j,k) data arrays, respectively. The ijk indices of these arrays correspond to elemental indices.
2. The stress components stored are tensor stress.

TIMER (Timing) Subcommand
CALC (Calculate) Input Group **CALC-TIMER**

Purpose: To track time required for various execution options.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **CALC** Group: None.

PCOM	SCOM	input parameters
	TIMER	*option*

option = (C) ON or OFF. Default = ON

Notes: 1. This option creates data manager arrays TIMER and NTIMER. NTIMER holds the names of code options being timed. TIMER holds the accumulated wall clock time since TIMER was activated. This option is useful for identifying where models are spending their execution time.

TFRM (transform) Subcommand
CALC (calculate) Input Group

Purpose: To compute transformed field variables. Velocity, displacement, stress or strain may be transformed into an alternative coordinate system for display. The transform is performed when the command is input.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM SCOM input parameters

TFRM (rate) *option (datanames)* TO *axisname ibegin iend jbegin jend kbegin kend*

rate = (I) Number of timesteps between TFRM updates. Only performed once at current time if *rate* is not input.

option = (C) Transformation option. Any of:

DISP=	displacement
VEL=	velocity
STRS=	stress
STRN=	strain
CORD=	coordinates

No default.

datanames = (C) Optional data array name input. If not transforming standard named data arrays, the user can provide a list of the data arrays containing the x-, y- and z- components of the velocity or displacement field or the xx, yy, zz, xy, yz, and xz components of the stress or strain fields, in the order listed.
No default.

axisname = (C) Name of an alternative coordinate system as defined by the **AXIS** primary command. Global quantities will be transformed into a local Cartesian system whose coordinate axes correspond to this coordinate system. No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the region of the model to transform.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the region of the model to transform.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the region of the model to transform.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. The transformed data computed by the **TFRM** subcommand are stored in their own data arrays. Storing this information can require a significant amount of memory. Transformed displacement data is stored in the D1TF(i,j,k), D2TF(i,j,k) and D3TF(i,j,k) arrays, where the ijk indices are nodal indices for the model and the D1TF array contains the local 1 direction component of the displacement field, the D2TF array contains the local 2 direction component, etc. Similarly, transformed velocity data is stored in the V1TF(i,j,k), V2TF(i,j,k) and V3TF(i,j,k) arrays. Transformed stress field components are stored in the S1TF(i,j,k), S2TF(i,j,k), S3TF(i,j,k), S4TF(i,j,k), S5TF(i,j,k), S6TF(i,j,k), where the ijk indices are elemental indices for the model and the component indicators: 1,2,3,4,5 and 6 correspond to the $\sigma 11$, $\sigma 22$, $\sigma 33$, $\sigma 12$, $\sigma 23$ and $\sigma 13$ stress components, respectively. Similarly, transformed strain field components are stored in the E1TF(i,j,k), E2TF(i,j,k), E3TF(i,j,k), E4TF(i,j,k), E5TF(i,j,k) and E6TF(i,j,k) data arrays.

2. When transforming field quantities to alternative coordinate systems (see **AXIS DEFN** subcommand), local directions 1, 2 and 3 are defined as:

Cartesian systems:	Direction 1 = local x' direction Direction 2 = local y' direction Direction 3 = local z' direction
Cylindrical systems:	Direction 1 = local radial direction Direction 2 = local θ (tangential) direction Direction 3 = local z' (axial) direction
Spherical systems:	Direction 1 = local radial direction Direction 2 = local θ direction Direction 3 = local ϕ direction

3. As the transformation is performed only on the region of the model defined by the indices input parameters, different transform systems may be used in different regions of the model. In the case of overlapping transform request, the last transform applies.

Examples:

```
calc
C   transform velocity field for the entire model into the cylindrical system name CYLA
        trfm vel to cyla
C   transform the stress field for a local region of the model into the spherical system
C   named SPHB
        trfm strs to sphb 1 20 1 20 10 30
        end
C       plot the radial component of the transformed stress field
grph plot s1tf
```

VELM (velocity magnitude) Subcommand
CALC (calculate) Input Group **CALC-VELM**

Purpose: To request that the code compute the magnitude of the velocity vector during the computation of the model's response.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	VELM	

Note: The VELM(I,j,k) data array will be created when this option is used. It is a nodal array and always contains the current magnitude value of the velocity vector at each node.

VMAG (vector magnitude) Subcommand
CALC (calculate) Input Group **CALC-VMAG**

Purpose: To compute the vector magnitude of a 2D or 3D set of vectors.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

VMAG *vmagname xdataname ydataname (zdataname)*

vmagname = (C) Name of the data array which will contain the vector magnitude values.
No default.

xdataname = (C) Name of the data array which contains the x-values for the set of vectors.
No default.

ydataname = (C) Name of the data array which contains the y-values for the set of vectors.
No default.

zdataname = (C) Optional input. Name of the data array which contains the z-values for the set of vectors. If not input, then a set of 2D vectors is assumed.
No default.

Note: This command is executed only when issued, not throughout the calculation.

VOLM (volume) Subcommand
CALC (calculate) Input Group **CALC-VOLM**

Purpose: To request that the volume of a group of continuum elements be calculated during the computation. The group of elements are defined by one or more ijk regions of the mesh.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **CALC** Group: None.

PCOM SCOM input parameters

VOLM *volumename (histname) ibegin iend jbegin jend kbegin kend*

volumename = (C) Name of this volume group. If *volumename* is the same as that defined on the preceeding **VOLM** subcommand, then this command will append the specified ijk region to those previously assigned to *volumename*. No default.

histname = (C) Optional parameter. The name of a previously defined DATA HIST curve. If specified a pressure associated with *volumename* will be calculated during the computation. This pressure can be applied to surfaces with the PLOD command. See Note 2. Default is no pressure calculated.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the region for volume computation.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the region for volume computation.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the region for volume computation.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. The VOLE(i) data array will be created when this option is used. It is a one dimensional array that contains the volume for each *volumename* defined (in the order they are defined) during a computation.

2. Specifying *histname* calculates a uniform fluid pressure associated with *volumename*. The history curve contains pairs of engineering volumetric strain vs.. pressure. Strains are considered to be positive in compression. Pressures are stored in the VOLP(i) data array, where i is one to the number of volumes defined.

VOLMINTG (volume integral) Subcommand
CALC (calculate) Input Group **CALC-VOLMINTG**

Purpose: To calculate the volume integral of an element array over a specified window.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CALC** Group: None.

PCOM SCOM input parameters

VOLMINTG *(rate) name array ibeg iend jbeg jend kbeg kend*

rate = (I) Optional. If entered the volume integral is calculated every *rate* timesteps. If rate is not entered the volume integral is calculated *only once at the current timestep.* See Note 1.

name = (C) Name identifying the volume integral.

array = (C) Element data array name to be integrated.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the interval.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the interval.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the interval.
Default: *kbegin* = 1, *kend* = maximum K

Notes:
1. If the *rate* parameter was input to allow recording the time history of the volume integrated array, the **VOLMINTG** command should be input before the **PRCS** command. To recalculate the volume integral every timestep, *rate* = 1 must be entered. To calculate the volume integral only once at a specific time during the simulation, the model's response should be computed up to desired time and then the **VOLMINTG** command input without the *rate* parameter.
2. Multiple **VOLMINTG** subcommands with the same *name* can be entered to associate multiple ijk windows with the same *name*. The resulting integral is the sum of the integrals over each ijk range. The values of *rate* and *array* are taken from the last entry for a given *name*.
3. Integral values are stored in data manager array CLVR(*n*) in the order input where *n* is the number of *names*. *name* and *array* are stored in order in data manager array CLVN(2,*n*).

VOLR (volumetric strain rate) Subcommand
CALC (calculate) Input Group **CALC-VOLR**

Purpose: To request that the volumetric strain rate be computed during the analysis. This will occur automatically if artificial viscosity is active in the analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM	SCOM	input parameters
	VOLR	-none-

Note: The volumetric strain rate data values are stored in the VOLR(i,j,k) data array. The ijk indices for this array corresponds to elemental indices.

VPHASE (phase velocity) Subcommand
CALC (calculate) Input Group **CALC-VPHASE**

Purpose: To fill an array with phase velocity vs. frequency values derived from an array containing wave number vs. frequency data.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **CALC** Group: None.

PCOM SCOM input parameters

VPHASE *datanamefrom datanameto (fmin fmax nf vmin vmax nv)*

datanamefrom = (C) The Data Manager array containing wave number vs. frequency data. The array must be data manager type 7 (mapped to coordinate vectors). This array will not be modified by this subcommand. No default.

datanameto = (C) The Data Manager array to be filled with phase velocity data. A data manager type 7 array must be previously set-up to accept the data unless the optional parameters are also specified. See note 1. No default.

fmin, fmax = (F) Minimum and maximum frequency range for new phase velocity array. Required if *datanameto* does not exist. Not used if *datanameto* already exists. See note 2. No default.

vmin, vmax = (F) Minimum and maximum velocity range for new phase velocity array. Required if *datanameto* does not exist. Not used if *datanameto* already exists. See note 2. No default.

nf, nv = (I) Number of frequency values and number of velocity values in new phase velocity array. Required if *datanameto* does not exist. Not used if *datanameto* already exists. See note 2. No default.

VPHASE (phase velocity) Subcommand
CALC (calculate) Input Group **CALC-VPHASE**

Notes: 1. The complete command sequence to create a phase velocity array from an existing wave number array is for example:

DATA OPEN *phasevel 2 $idim $jdim * f*

MGR STRUCTURE *phasevel freq frequency $frangmin $frangmax &*
pvel phase_velocity $vrangmin $vrangmax

VPHASE *wavearray phasevel*
END

2. Alternatively to the above example, the user can allow the program to open and set the structure of *phasevel* automatically if it does not already exist by specifying the frequency and velocity ranges and the size of the array using the optional parameters. Default names derived from *phasevel* are used for the axis labels. The following example will achieve the same result as the example in note 1:

DATA VPHASE *wavearray phasevel $frangmin $frangmax$idim &*
$vrangmin $vrangmax $jdim

CIRC (circuit) Input Command

Purpose: To define circuits.

Use Is: Optional.

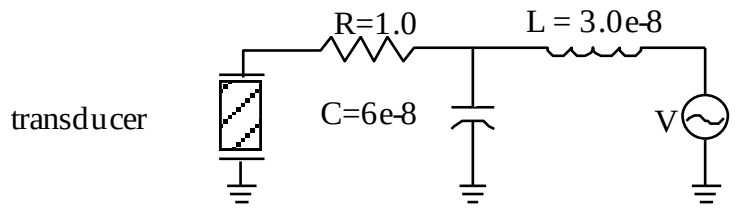
Multiple Use: Allowed.

Order Dependence: Must precede **PIEZ CONN** subcommands which reference this circuit.

PCOM	SCOM	input parameters
CIRC		
	DEFN	*circname*
	ELEM	*type connect value(s)*
	DSRC	*qnsrc qnref nodsrc nodref scale*
	PRNT	
	END	

Notes: 1. The following input commands demonstrate the definition of a simple matching circuit named MACH. The response of the transducer will be modeled by the continuum finite element representation of the piezoelectric material. The electrodes on the top and bottom of the piezoelectric material must be defined using the **PIEZ DEFN** subcommand. The **PIEZ CONN** subcommand will connect the MACH circuit to the top electrode and define the driving source for the circuit. The **PIEZ BC** subcommand is used to ground the bottom electrode.

```
CIRC
     DEFN MACH
     ELEM REST SERS 1.0
     ELEM CPAC SHNT 6.0e-8
     ELEM INDR SERS 3.e-8
```



Notes: 2. Circuits are connected to electrodes in the **PIEZ** command group by using the **CONN** subcommand. For example the following input set might be used to connect the example circuit shown above to the electrode named top.

```
PIEZ
    WNDO 1 21 1 101     /* electric window
    DEFN TOP $SCALE     /* define top electrode
    NODE 1 1 1 101
    DEFN BOTM $SCALE    /* define bottom electrode
    NODE 21 21 1 101
    CONN TOP MACH VOLT FUNC 1.      /* connect circuit MACH to top
    BC BOTM GRND        /* ground bottom electrode
    END
```

3. Often times, the **PZFlex** model might use a 2Dimensional approximation of a 3Dimensional model or it might use symmetry boundary conditions to allow modeling only 1/2 or 1/4 of the physical structure. These modeling assumptions affect the charge that is computed on the electrodes within the model and, consequently, affect the response of circuits connected to the model's electrodes. The $SCALE parameter shown on the **DEFN** subcommands in the above example is used to scale the charge on the model's electrodes to the appropriate value for use with the actual circuit parameters defined by the **CIRC** command group. Proper scaling of the electrode's charge is **critical** to achieve proper circuit response.

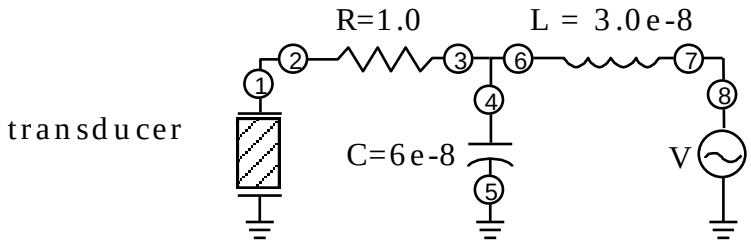
4. Circuit voltage and charge data are stored in the CRTV(i) and CRTQ(i) data arrays. These arrays are one dimensional arrays whose indices represent a specific degree-of-freedom (unknown) for the group of circuits connected to the model electrodes. The code prints a table identifying which degree-of-freedom is at which point in the circuit. For example, the input in (2) above will produce the following print out in the flxprt file:

```
Electrode   1 name=top  number of nodes=  101
Electrode   2 name=botm number of nodes=  101

Indices of circuit unknowns in crtq and crtv arrays
-------- -- ------- -------- -- ---- -- ---- ------
Electrode top          1
sers rest  1.0000E+00   2   3
shnt cpac  6.0000E-08   4   5
sers indc  3.0000E-08   6   7
Source                 8
```

This table shows that there are 8 degrees-of freedom (dof) for the circuit solution matrix for this problem. They correlate with the locations shown in the circuit diagram below. Note that, except for the electrode and the source, there are two dofs representing the circuit state on both sides of each circuit component. Often, there is more than one dof at the same physical location in the circuit. For example, dofs 1 and 2 are at the same location and likewise for dofs 3, 4 and 6. At junctions like these, the voltages are equal and the charges sum to zero. This means that for a junction of two dofs, the two charge values will have the same magnitude but different signs.

Notes: 4 (continued)



Location of circuit degrees of freedom for the above example

5. To request a time history of voltage and charge for a point in the circuit between the inductor and the source, the following commands may be used. These commands request two time histories, voltage and charge, which occur between the inductor and the voltage source.

```
POUT
    HIST CRTV 7 7 1
    HIST CRTQ 7 7 1
    END
```

DEFN (definition) Subcommand
CIRC (circuit) Input Group **CIRC-DEFN**

Purpose: To initiate a circuit definition.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **CIRC** Group: Must precede **ELEM** subcommands which define this circuit.

PCOM	SCOM	input parameters
	DEFN	*circname*

circname = (C) Unique name assigned to this circuit.

ELEM (element) Subcommand
CIRC (circuit) Input Group **CIRC-ELEM**

Purpose: To define the elements of a circuit. Resistors, capacitors, inductors, ideal transformers and a cable model are currently implemented. The circuit elements are specified in order proceeding from the electrode to the source.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **ELEM** Group: Must follow **DEFN** subcommand which initated this circuit.

PCOM SCOM _____ input parameters

ELEM *type connect value(s)*

type = (C) Type of circuit component.
Any of: REST = resistor
CPAC = capacitor
INDR = inductor
ITFR = ideal transformer
COAX = coaxial cable
SRLC = series RLC compound element
PRLC = parallel RLC compound element

connect = (C) Connectivity.
Any of: SERS = in series
SHNT = shunt to ground

values = (F) One or more values defining the properties of this component. See below for specifics.

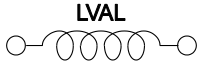
ELEM REST *connect rval* for a resistor

rval= (F) Resistance. Default = 0.0

ELEM CPAC *connect cval* for a capacitor

cval= (F) Capacitance. Default = 0.0

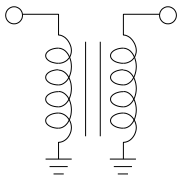
ELEM INDR *connect lval* for an inductor



lval= (F) Inductance. Default = 0.0

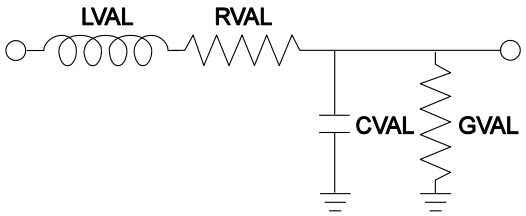
PCOM SCOM ——— input parameters ———

ELEM ITFR *connect trat* for an ideal transformer



trat= (F) Ratio of turns on the primary (transducer side) to turns on the secondary. Default = 1.0

ELEM COAX *connect lval rval cval gval length nelem* for a coaxial cable



lval= (F) Series inductance per unit length Default = 0.0

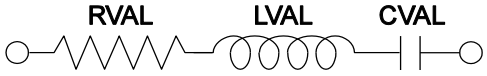
rval= (F) Series resistance per unit length Default = 0.0

cval= (F) Shunt capacitance per unit length Default = 0.0

gval= (F) Shunt conductance per unit length Default = 0.0

length= (F) Cable length Default = 0.0

nelem= (I) Number of discrete elements used to approximate cable. Default = 1. (see note 1 below)



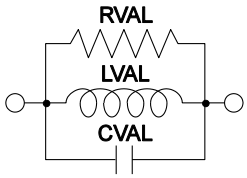
ELEM SRLC *connect rval lval cval* for a series RLC element

rval= (F) Series inductance Default = 0.0, see note 4 below.

lval= (F) Series inductance Default = 0.0

cval= (F) Series capacitance Default = 0.0

ELEM PRLC *connect rval lval cval* for a parallel RLC element

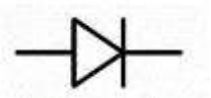


rval= (F) Parallel inductance Default = 0.0, see note 5 below.

lval= (F) Parallel inductance Default = 0.0

cval= (F) Parallel capacitance Default = 0.0

ELEM DIOD *connect vicurv direction* for a DIODE element



vicurv = (C) Name of previously defined voltage vs. current curve. See note 6.

direction= (C) Direction. FWD (forward) or REV(reverse). Default = FWD, passes current.

Notes: 1. A discrete cable model is used. For good accuracy, *nelem* should be set to (20 * *l* * *fmax* / *v*) where *fmax* is the highest frequency to be resolved, *v* is the signal propagation speed in the cable, and *l* is the cable length. For example, if *fmax* = 10 Mhz, *l* = 3 m, *v* = 2 x 10^8 m/sec, then *nelem* = 3 should be used.

2. For low-loss cables where shunt conductance is neglected, the cable model parameters may be approximated as follows:
   lval = *Z0* / *v*
   cval = *1.0* / (*Z0* * *v*)
   rval = *0.2303* * *Z0* * *loss*
   gval = *0.*
   where *v* is the signal propagation speed in the cable (typically 0.5 to 0.7 time the speed of light in a vacuum), *Z0* is the characteristic impedance of the cable, and *loss* is the signal attenuation in dB/unit length.

3. See *Fields and Waves in Communication Electronics* by Ramo, Whinnery and Van Duzer, pp. 245 - 251, for further discussion.

4. The SRLC compound element consists of a resistor, an inductor, and a capacitor connected in series. It may be connected to the circuit in either series or shunt to ground. Setting *cval* = 0 or default results in infinite capacitance (ie, no capacitor).

5. Likewise, the PRLC compound element consists of a resistor, an inducto, and a capacitor connected in parallel. Setting *rval* = 0 or default results in infinite resistance (i.e., no resistor). Setting *lval* = 0 results in infinite inductance (i.e., no inductor).

6. The curve is defined as voltage difference in first column vs. current in the second column. 20 points max. The input curve should be monotonic, with no infinite slopes. Note that DIODES are nonlinear and will therefore trigger a more CPU intensive circuit solver.

DSRC (Dependent Source) Subcommand
CIRC (Circuit) Input Group **CIRC-DSRC**

Purpose: To define a dependent source. This impleents a linear relation between the voltage or current at the previously grounded end of a shunt connected element and the voltage or current at some other point in the circuit.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **ELEM** Group: Must follow **DEFN** subcommand which initated this circuit.

PCOM SCOM input parameters

DSRC *qnsrc qnref nodsrc nodref scale*

qnsrc = (C) Quantity specifed at source node.
Any of: VOLT = voltage
CRNT = current

qnref = (C) Quantity specifed at reference node.
Any of: VOLT = voltage
CRNT = current

nodsrc = (I) Circuit node where dependent source is specified. Must be the previously grounded end of a shunt connected element.

nodref = (I) Circuit node on which the source depends. See Note 1.

scale = (F) Scale factor relating source and reference quantities. See Notes 2 and 3.

Notes: 1. The circuit nodes are numbered starting at the electrode. 1 is the transducer, 2 & 3 are the first circuit element, etc. The numbering is echoed to the flxprt file at the end of the PIEZ input section. The electrode cannot be used as a reference node. If this is needed, insert a null resistor and use it as the reference.
2. For example, to specify that the voltage at node 3 is -0.3 times the current at node 5:
DSRC volt crnt 3 5 -0.3
3. Note: You cannot use this to connect between two separate circuits, must be in one DEFN.

PRNT (print) Subcommand
CIRC (circuit) Input Group **CIRC-PRNT**

Purpose: To print circuit information.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **CIRC** Group: Should follow the input commands which define the information that the user desires to print.

PCOM	SCOM	input parameters
	PRNT	(none)

DATA Input Command

DATA

Purpose: To provide the user with direct IO access to data manager arrays.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Should be used at the appropriate place in the input stream to read in or write out the data of interest.

PCOM SCOM input parameters

DATA

ABS	*datanameabs dataname1 dataname2 . . . datanamen*
ADD	*datanamefrom scale datanameto ibegin iend jbegin jend kbegin kend*
CDDO	*dataname filename ibegin iend jbegin jend kbegin kend*
CLOS	*dataname*
COPY	*datanamefrom datanameto*
CPYG	*datanamefrom ibgfrom iincfrom jbgfrom jincfrom kbgfrom kincfrom &*
	datanameto ibgto iendto jbgto jendto kbgto kendto
DIVIDE	*dataname_denom datamane_numer*
EXPAND	*array arraynew*
EXPONENT	*dataname exponent*
FFT	*realarray imagarray direction inverse*
FFTSHFT	*array1 (array2 array3 …arrayN) direction*
FILE	*option filename (type)*
FORM	*option format idtag*
HIST	*dataname nvalues filename scaletime scaledata*
IMPORT	THRM *filename matname*
IN	*dataname*
INTERP	*array arraynew*
INTEXT	*dataname type filename ncolbegin ncolend nskip*
LOG	*dataname logdataname maxvalue*
MATH	*arraynew* = [algebraic expression involving arrays]
MGR	DIM *dataname ndimension ndim1 (ndim2) (ndim3)*
MGR	MAP *dataname itype mapname*

List of **DATA** commands continues

PCOM	SCOM	input parameters
DATA		
	MIRR	*array arraynew axis option sidopt ibeg iend jbeg jend kbeg kend*
	MULTIPLY	*dataname1 dataname2*
	OPEN	*dataname ndimension ndimi ndimj ndimk datatype*
	OUT	*dataname*
	OUT0	*dataname*
	OUT1	*dataname*
	PHAD	*ampfrom phasfrom scale ampto phasto ibegin iend jbegin jend kbegin kend*
	POLR	*real imag option amp phas ibegin iend jbegin jend kbegin kend*
	REMAP	*pathname source_array destination_array axis nsample &*
		ibegin iend jbegin jend kbegin kend
	SCAL	*dataname scalevalue ibegin iend jbegin jend kbegin kend*
	TABL	*dataname idim jdim*
	VPHASE	*datanamefrom datanameto*
	END	

ABS (absolute value) Subcommand
DATA Input Group**DATA-ABS**

Purpose: To generate a data array containing the maximum absolute values from one or more data arrays.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM	SCOM	input parameters
	ABS	*datanameabs datataname1 dataname2 . . . datanamen*

datanameabs = (C) Data Manager array containing the maximum absolute values. This data array is created by this subcommand and is the same size as *dataname1*.
No default.

datanamen = (C) List of data names from which to determine the maximum absolute value.
No default.

Note: The action of this subcommand is equivalent to the following logic.for each array value.
datanameabs(i,j,k)= max[|*dataname1*(i,j,k)|, |*dataname*2(i,j,k)|, . . ., |*datanamen*(i,j,k)|]

ADD Subcommand
DATA Input Group**DATA-ADD**

Purpose: To add the values stored in one Data Manager data group, scaled by some scale factor, to the values of another data group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM	SCOM	input parameters
	ADD	*datanamefrom scale datanameto ibegin iend jbegin jend kbegin kend*

datanamefrom = (C) Data Manager array containing the values that are going to be scaled and added. This data array is not affected by this subcommand.
No default.

scale = (F) Scale factor to scale the values in *datanamefrom* before adding them to *datanameto*.
Default = 0.0

datanameto = (C) Data Manager array to add values to. The array is modified by this subcommand.
No default.

ibegin, iend = (I) Beginning and ending I-indices of data array locations that are modified by this subcommand.
Default: *ibegin* = 1, *iend* = maximum I of *datanameto*

jbegin, jend = (I) Beginning and ending J-indices of data array locations that are modified by this subcommand.
Default: *jbegin* = 1, *jend* = maximum J of *datanameto*

kbegin, kend = (I) Beginning and ending K-indices of data array locations that are modified by this subcommand.
Default: *kbegin* = 1, *kend* = maximum K of *datanameto*

Note: To combine data arrays with the **ADD** subcommand without actually changing the values stored in *datanameto*, a working copy of the array may be created using the **COPY** subcommand which can then be used by the **ADD** subcommand for the desired combination.

CDDO Subcommand
DATA Input Group**DATA-CDDO**

Purpose: To write values contained within a Data Manager array to a disk file in a comma-delimited format that can be imported into EXCEL or some other program. This command is typically used to output a 2D section of a 2D or 3D array. It will also write out a 3D region if an array is requested.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM	SCOM	input parameters
	CDDO	*dataname filename ibegin iend jbegin jend kbegin kend*

dataname = (C) Data Manager array name.
No default.

filename = (C) Name of a file where the data is to be written..
No default.

ibegin, iend = (I) Beginning and ending I-indices of tha array section to be written.
Default: *ibegin*=1, *iend*=maximum I

jbegin, jend = (I) Beginning and ending J-indices of the array section to be written.
Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of the array section to be written.
Default: *kbegin*=1, *kend*=maximum K

Notes:
1. If the array is 3D, the specified indices should form a 2D section of the array if the data is to be imported into a program such as EXCEL.
2. If the data being written has a constant I-index, i.e., only the J- and K- indices vary, then each line of data written to the file has a constant K-index and the J-index varies along the line. For a constant J-index section, each line of data is for a constant K-index. For a constant K-index section, each row of data is for a constant J-index.
3. If a 3D array (or subregion of the array) is written to a file, each line of data has a constant J- and K-index, and the I-index varies along the line.
The first line of data will be for J = *jbegin*, K = *kbegin*. The second line is for
J = *jbegin*+1, K= *kbegin*, etc. I.e., the J-index permutates first and then the K-index.

CLOS Subcommand
DATA Input Group

DATA-CLOS

Purpose: To close and delete a data array within the Data Manager.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

CLOS *dataname*

dataname = (C) Name of the Data Manager array to close.
No default. Entering the keyword "all" completely clesr the data manager.

COPY Subcommand
DATA Input Group **DATA-COPY**

Purpose: To create a duplicate copy of a Data Manager array with a different name.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

COPY *datanamefrom datanameto*

datanamefrom = (C) Data Manager array to copy. The array is not be modified by this subcommand.
No default.

datanameto = (C) Duplicate Data Manager array name. If this data array previously existed, it is replaced by the duplicate copy.
No default.

CPYG (generalized copy) Subcommand
DATA Input Group **DATA-CPYG**

Purpose: To copy a portion of a Data Manager array into another Data Manager array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

CPYG *datanamefrom ibgfrom iincfrom jbgfrom jincfrom kbgfrom kincfrom &*
datanameto ibgto iendto jbgto jendto kbgto kendto

datanamefrom = (C) Data Manager array to copy. The array is not modified by this subcommand. No default.

ibgfrom, jbgfrom, kbgfrom = (I) Beginning i, j, and k indicies of *datanamefrom*. No default.

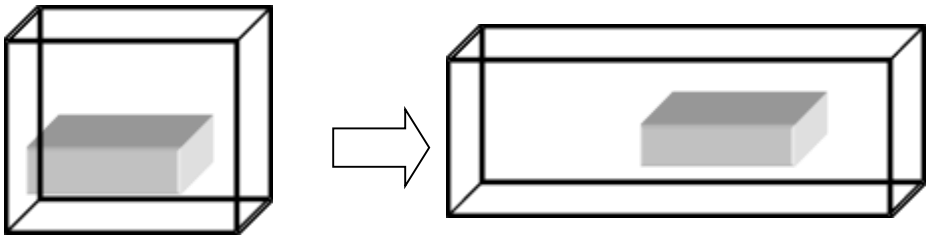
iincfrom, jincfrom, kincfrom = (I) i, j, and k index increments of *datanamefrom*. No default.

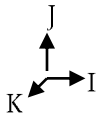
datanameto = (C) Duplicate Data Manager array name. Must already exist. No default.

ibgto, jbgto, kbgto = (I) Beginning i, j, and k indicies of *datanameto*. Default = 1.

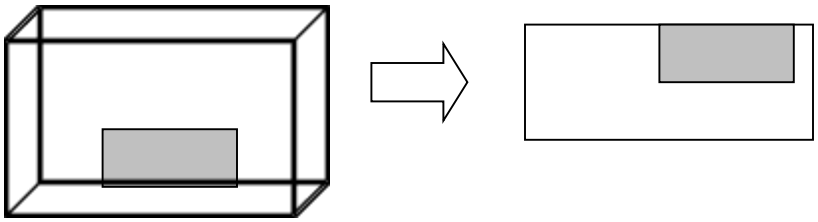
iendto, jendto, kendto = (I) Ending i, j, and k index indicies of *datanameto*. Default = maximum i,j,k respectively.

DATA-CPYG

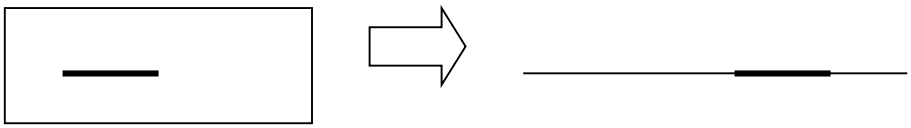




Copy IJK region of 3-D array to equivalent IJK region of another 3-D array



Copy I-J plane of 3-D array to equivalaent I-J region of a 2-D array



Copy I-line of data from a 2-D (or 3-D array) to an equivalent I-region of a 1-D array

Notes: 1. The number of dimensions of *datanameto* and *datanamefrom* do not have to match. However, the command is designed to presereve the IJK orientation of the data during the copy process. Hence, only an I-J plane of data can be copied into a 2-d array and only an I-line of data can be copied into a 1-D array.

DIVIDE Subcommand
DATA Input Group **DATA-DIVIDE**

Purpose: To divide the individual members of an array in the Data Manager by those contained in an identically sized array. The results will replace the original values contained in the numerator array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: Must follow the point at which the arrays have been created.

PCOM	SCOM	input parameters
	DIVIDE	*dataname_denom datamane_numer*

dataname_denom= (C) Name (including path) of the data array that will be the denominator for the divide operation. The denominator array will remain unchanged by this command.
No default.

dataname_numer= (C) Name (including path) of the data array that is the numerator for the divide operation. The results of the divide operation will replace the original content of the numerator array.
No default.

Note: Example: The following example redefines each member of array numer as follows:

numer(*i*) = numer(*i*) / denom(*i*)

```
DATA
    DIVIDE denom numer
    END
```

EXPAND Subcommand
DATA Input Group **DATA-EXPAND**

Purpose: To copy a mapped (ijk windows) array into a full ijk array. This can be useful for plotting or data manipulation.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM input parameters

EXPAND *array arraynew*

array = (C) Name of an existing mapped array.

arraynew = (C) Name of full array.

Notes: 1. *array must be an existing mapped array.*
2. *If arraynew* exists, it must be the correct size. It will be created if it does not exist.

EXPONENT Subcommand
DATA Input Group

DATA-EXPONENT

Purpose: To compute the value of x^p for each member of an array in the Data Manager, i.e., each member of the array is raised to the exponent p.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: Must follow the point at which the array has been created.

PCOM SCOM input parameters

EXPONENT *dataname exponent*

dataname = (C) Name (including path) of the array to be inverted. Note that this opertation inverts each individual member of the array, i.e., computes the reciprocal of each term. This command changes the values contained in the data array. No default.

exponent = (X) Exponent to use. May be either an integer or real valued number. Default = 0

Note: Any request that results in a "divide by 0," i.e., taking the -1 power of a 0.0 entry, is ignored and the original data value remains unchanged.

The following example initially replaces the value of each member of "data" with its square. The next command replaces each entry with the reciprocal of the square of the original entry.

```
DATA
    EXPONENT data  2    /* data(i) = data(i) 2
    EXPONENT data  -1   /* data(i) = 1. / data(i)
    END
```

FFT Subcommand
DATA Input Group

DATA-FFT

Purpose: To perform an FFT or inverse FFT on an array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM	SCOM	input parameters
	FFT	*realarray (imagarray) (direction) inverse newrealarray (newimagarray)*

realarray = (C) Name of the real part of the array. No Default.

imagarray = (C) Optional. Name of the imaginary part of the array. Default = *realarray*_imag. If *imagarray* does not already exist, it is created and filled with zeros.

direction = (C) Optional. Indicates direction of the FFT. I, J, or K. Default = I.

inverse = (C) If set to INV an inverse FFT is performed on the array. Default is to perform forward FFT.

new*realarray* = (C) Name of the real part of the new array. If not defaulted, *newrealarray* is created as a copy of *realarray* and the FFT is performed only on *newrealarray*.

new*imagarray* = (C) Optional. Name of the imaginary part of the new array. Default = new*realarray*_imag. If *newimagarray* does not already exist, it is created and filled with zeros.

Notes: 1. If a *newrealarray* name is entered, the FFT is performed only on the copy of *realarray* and *realarray* is left unchanged.
2. 2D FFTs can be performed by successive calls of DATA FFT on the same array with different *directions*.

FFTSHFT Subcommand
DATA Input Group

DATA-FFTSHFT

Purpose: To shift values in an array by half of the array length in a given direction along the frequency axis after the array has undergone an FFT. This operation allows the large amplitude frequencies to be centered.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM SCOM input parameters

FFTSHFT *array1 (array2 array3 …arrayN) direction*

arrayN = (C) Names of the arrays to be shifted. At least one array name must be entered. No Default.

direction = (C) Indicates direction of the shift. I, J, or K. Default = I.

Notes: 1. Values in the array are shifted by half of the array length in the given direction so that the first half of the array becomes the second half and vice versa.
2. If the array is not of even length in the shift direction, a warning is posted but the shift is performed, i.e., there is a column at the end of the array that is not shifted.
3. The array must be at least of length 2 in the shift direction.

FILE Subcommand
DATA Input Group **DATA-FILE**

Purpose: To assign an external file as the current Data Input File or Data Output File.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

FILE *option filename (type)*

option = (C) Data file option. Any of: IN - defines Data Input File
OUT - defines Data Output File

filename = (C) Name of remote file.
(filename may be up to 80 characters).
No default.

type= (C) enter MAT for data and header files in .mat (MATLAB) format, BIN for ASCII header file and raw binary file (e.g. written by C or other non-FORTRAN code). Default is standard FLEX FORTRAN format.

Notes: 1. If the **FILE** subcommand is used, any file previously assigned as the Data Input or Output File (depending on the *option* specified) will be closed before the new file is opened.
2. Assigning a file name which is the same as the current file is equivalent to repositioning the read position to the beginning of the file.
3. For type=BIN, there should be a binary file *filename.bin* , and an ASCII file *filename.hdr.* The filename.hdr file is an ASCII file that stores the arrays names/dimensions/precision 'data manager' style:
name #dimensions dim1 dim2 dim3 type
where type = any of: float32, float64, int32, int64
filename.bin stores the arrays contents in binary format/Fortran column order.
Multiple arrays per file are supported.

For example:
test.hdr
```
pres 3 1333 500 1 float32
```
test.bin is a binary file stores 1333x500 32 bit floating point numbers ordered by column.

FORM Subcommand
DATA Input Group

DATA-FORM

Purpose: To assign the format for reading/writing external Data Input Files or Data Output Files. This format applies to Output Files written with

Use Is: Optional.

Multiple Use: Not Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM ____ input parameters ____

FORM *option format idtag mode*

option = (C) Data file option. Any of: IN - defines Data Input File
OUT - defines Data Output File

format = (C) Format of external file. Any of:
MATLAB — use matlab .mat file format
No default.

idtag = (C) Two letter identifier for arrays written to matlab .mat files.
Default = D1.

mode = (C) Any of w, w7.3. Default = w7.3.

Notes: 1. If this subcommand is not used standard Data Input and Data Output files will be read and written when the OUT subcommand is issued.
2. Currently the only output format option is MATLAB. Once output is written in either Flex or matlab format, the same format should be used for all subsequent DATA OUT subcommands.
3. The 2-letter *idtag* will be used as a prefix for all arrays written to matlab .mat files. All arrays written to .mat files also have a 7-digit timestep prefix to identify the timestep at which the array was written. For example: d1_0001234_pres
4. w = MATLAB 7.0 format; w7.3 = Creates a MAT-file in an HDF5-based format that can store objects that occupy more than 2 GB
5. Example input for writing to a matlab .mat file:

```
DATA
    FILE  out  filename.mat
    FORM out matlab d1
    OUT modl
    OUT pres
    END
```

HIST (History) Subcommand
DATA Input Group **DATA-HIST**

Purpose: To provide a simple way of inputting digitized time history data into a user-specified Data Manager array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM SCOM input parameters

HIST *dataname nvalues filename scaletime scaledata shifttime (option)*

dataname = (C) Data Manager array name.
No default.

nvalues = (I) Number of digitized time history values.
Default: If *filename* is specified, the user may default *nvalues* and the program will read information from filename until the first blank line is encountered or the end-of-file is detecte.

filename = (C) Name of a file containing the time history values. Default is to read the time history values immediately following the **DATA HIST** command from the Job Input File

scaletime = (F) Scale factor to scale the time values by.
Default = 1.0

scaledata = (F) Scale factor to scale the data values by.
Default = 1.0

shifttime = (F) Timeshift value to add to the time values in *filename*.
Default = 0.0

option = (F) Optional parameter to control checking for monotonically increasing time. If not input, the time information for this time history must be monotonically increasing. If set to NCEQ, then duplicate time values are allowed but time values that are less than preceding time entries are not allowed. If set to NCLE, then time values are not checked for monotonic increasing values. Note that allowing non-monotonic data for a loading function may cause difficulties when the time history data for a value at a specified time are interpolated.
No default

Note: The format of the time history data is: one pair (time, data value) per record for *nvalues* records. The data must be in free-format text form.

Example 1:
```
DATA HIST  H1  3          /* read in time history H1 from Job Input File
           0.0      0.0   /* one data pair is specified on each line, time is input first
           1.0      10.0
           5.0      0.0
```

Example 2:
```
DATA HIST  H3  3  MY.HIST.FILE        /* read in time history H3 from another file

< this data must be placed in file: MY.HIST.FILE >
           0.0      0.0   /* one data pair is specified on each line, time is input first
     1.0      10.0
     5.0      0.0
```

IMPORT Subcommand
DATA Input Group

DATA-IMPORT

Purpose: To read a file (OUT1 FORMAT) containing an option-specific set of arrays.
Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

IMPORT *THRM filename matname (subname)*

filename = (C) Name of the file containing the data.
No default.

matname = (C) Material name for thermal data..
No default

subname = (C) Name of subdirectory where data is to be placed..
Default: *matname.*

Note: IMPORT THRM provides a convenient means of importing tables of specific heat and thermal conductivity vs.. temperature and/or other variable. The file should contain in the following order: *tmpr:matname*. *vrat:matname*, *c:matname*, *kx:matname*, *ky:matname*, and for 3D models; *kz:matname*. *c, kx, ky,* and *kz* are dimensioned (*nt, nv*) where *nt* is the number of temperatures and *nv* is the number of entries in *vrat*. *Tmpr* and *vrat* should be in motonically increasing order. If there is no dependency on either *tmpr* or *vrat*, enter an array with a single value.

IN Subcommand
DATA Input Group

DATA-IN

Purpose: To assign data values stored on an external file to an internal Data Manager array. The array may or may not already exist. The data is read from the Data Input File.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM	SCOM	input parameters
	IN	*dataname (option)*

dataname = (C) Data Manager array name that contains the data once it has been imported from the Data Input File.
If not input, the array name will be set to the array name contained on the Data Input File.

option = (C) Optional input parameter applicable only if importing a subregion of an array previously written out to the Data Out File. By default, the code reads in a subregion of an array and places these values within an array of the original size. To read in the partial array but place it in an array sized to match the subregion, specify PART as the *option* parameter.

Notes: 1. See the section "Files and Formats" for a description of the format of the Data Input File.
2. Each **IN** subcommand reads the next data array contained on the Data Input File, i.e., the third **IN** subcommand input inports the third data array saved in the data file being read.

INTERP (interpolation) Subcommand
DATA Input Group **DATA-INTERP**

Purpose: To interpolate a mapped array (type 7) to a new equally spaced mapped array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM	SCOM	input parameters
	INTERP	*array arraynew*

array = (C) Name of an existing array to interpolate from.

arraynew = (C) Name of existing data manager array to store interpolated values.

Notes: 1. Both *array* and *arraynew* must be existing arrays of type 7 (i.e., mapped to coordinate vectors). The DATA MGR STRUCTURE command can be used to generate an empty *arraynew* of the desired dimensions.
2. *array* and *arraynew* can be one-, two-, or three-dimensional arrays; they can be of different sizes but should have the same number of dimensions.

INTEXT Subcommand
DATA Input Group **DATA-INTEXT**

Purpose: To assign data values which are stored on an external file to an internal Data Manager array. The array may or may not already exist. The format of the external file is rows and columns of text.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM input parameters

INTEXT *dataname type filename ncolbegin ncolend nskip*

dataname = (C) The data manager array name that will contain the data once it has been imported from the Data Input File. No default.

type= (C) Any of: C for character data, I for integer data or F for floating point data. No default.

filename = (C) name of file to be read. No default.

ncolbegin, ncolend= (I) Beginning and ending columns of data to be read into dataname. Default is all columns.

nskip = (I) Number of lines to skip before reading data. Default = 0. See note 1.

Notes: 1. For Integer or Real types, column labels are automatically skipped.
2. The maximum amount of data read from a row is limited to no more than 300 entries and no more than 6000 characters.

LOG (logarithm) Subcommand
DATA Input Group**DATA-LOG**

Purpose: To compute a logarithmic representation of a data array for plotting purposes. This option can be useful to highlight large variations in field values.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: Must follow the point at which the data array is created.

PCOM SCOM input parameters

LOG *dataname logdataname maxvalue*

dataname = (C) Name of data array. No default.

logdataname = (C) Name of data array that will contain the log representation. No default.

maxvalue = (I) Maximum value used for computing logarithm. Default is maximum absolute value of *dataname*.

Notes: 1. This option produces a set of data contained in *logdataname* whose values range from -3.0 to +3.0. The value *maxvalue* corresponds to +3.0 and -*maxvalue* corresponds to -3.0. Three orders of magnitude reduction from maxvalue is mapped to the range 0.0 to 3.0. Array values less than the three order of magnitude limit are set to 0.0.
2. Negative data values are automatically accounted for by this option.

MATH Subcommand
DATA Input Group **DATA-MATH**

Purpose: To perform general math operations on arrays. This option is similar to the **MAKE** command in Review, except that it operates on arrays rather than time histories.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: Must follow the point at which the arrays in the algebraic expression have been created.

PCOM SCOM input parameters

MATH *arraynew* = [algebraic expression involving arrays]

arraynew = (X) Name (including path) of target data array. It will be created if it does not exist. It will be created to match the size and type of the first data array encountered in the algebraic expression. No default.

[algebraic expression] = (X) The arrays in the algebraic expression must exist, and must be of the same type (i.e., real or integer) and size.

arraynew is created based on a mathematical expression f(a1[...],a2[...], etc.) which may contain any of the Symbol math operators. Each a_n is a pre-existing data array identified by the construct:

{ *arrayname* }

where the bounding braces are required. All inputs in the expression must be blank-delimited.

Note: Example,

```
DATA
    MATH VMAG = SQRT ( { AMPX } * { AMPX } + { AMPY } * { AMPY } )
    END
```

This example computes the magnitude of a 2D vector field whose x-component and y-component values are strored in data arrays AMPX and AMPY, respectively. The magnitude value for each array location is evaulated and stored in the equivalent location in data array VMAG.

Note that this option executes much faster than SYMB #GET and #SET commands to perform the computations on each array element individually.

MGR (manager) Subcommand
DATA Input Group **DATA-MGR**

Purpose: To override or set the geometry mapping or dimensionality of a Data Manager array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: Must follow the point at which the data array is created.

PCOM SCOM input parameters

MGR DIM *dataname ndimension ndim1 (ndim2) (ndim3)*
or
MGR MAP *dataname itype mapname*
or
MGR STRUCTURE *dataname icrd ilabel rimn rimx &*
jcrd jlabel rjmn rjmx &
kcrd klabel rkmn rkmx

dataname = (C) Name of data array. No default.

ndimension = (I) Number of dimensions for the data array. From 1 to 3. No default.

ndim1 = (I) Size of the first dimension of the array. No default.

ndim2 = (I) Size of the second dimension of the array. Input only if *ndimension* > 1. No default.

ndim3 = (I) Size of the third dimension of the array. Input only if *ndimension* > 2. No default.

itype = (I) Type of geometry mapping for this data array.
Set to 11 if full nodal ijk mapping
Set to 21 for full elemental ijk mapping
Default = 0, not mapped to grid geometry.

mapname = (C) Name of data array containing mapping information for *dataname*. Default = no mapping data array.

icrd, jcrd, kcrd = (C) Names coordinate vectors for mapped arrays. If a coordinate vector name is defaulted, it is not set.

ilabel, jlabel, klabel = (C) Axis labels for mapped arrays.
Default for *ilabel* = 'time'
Default for *jlabel* = 'coord1'
Default for *klabel* = 'coord2'

PCOM SCOM input parameters

rimn, rimx, rjmn,
rjmx, rkmn, rkmx = (F) Minimum and maximum values for each coordinate vector.
Default to leave existing values if *icrd, jcrd, kcrd* exist.
Default for *rimn, rjmn, rkmn* = 0. if *icrd, jcrd, kcrd* do not exist.
Default for *rimx, rjmx, rkmx* = 1. if *icrd, jcrd, kcrd* do not exist.

MIRR Subcommand
DATA Input Group **DATA-MIRR**

Purpose: To mirror an existing Data Manager array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

MIRR *array arraynew axis option (sidopt) ibeg iend jbeg jend kbeg kend*

array = (C) Name of existing Data Manager array.

array = (C) Name of new array to hold mirrored data. No Default.

axis = (C) Mirror axis (2D) or plane (3D).
For 2D, any of: I — mirror about I-axis
J — mirror about J-axis
For 3D, any of: IJ — mirror about IJ plane
JK — mirror about JK plane
KI — mirror about KI plane
No Default.

option = (C) Symmetry option. Either SYMM or ASYMM. Default = SYMM.

sidopt = (C) Side option. Either of:
MIN: the data is mirrored about the minimum index
MAX: the data is mirrored about the maximum index
Default = MIN.

ibeg, iend = (I) Beginning and ending I-indices of data array locations in *array* to be mirrored and placed in *arraynew*.
Default: *ibegin* = 1, *iend* = maximum I of *array*

jbeg, jend = (I) Beginning and ending J-indices of data array locations in *array* to be mirrored and placed in *arraynew*.
Default: *jbegin* = 1, *jend* = maximum J of *array*

kbeg, kend = (I) Beginning and ending K-indices of data array locations in *array* to be mirrored and placed in *arraynew*.
Default: *kbegin* = 1, *kend* = maximum K of *array*

Notes: 1. 1D vectors can be treated as 2D arrays (n,1).
2. The data are always mirrored about the axis or the minimum index if bounding indices are specified.

MULTIPLY Subcommand
DATA Input Group **DATA-MULTIPLY**

Purpose: To multiply the individual values contained in two identically sized arrays contained in the Data Manager. The results replace the original values contained in the second array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: Must follow the point at which the arrays have been created.

PCOM SCOM input parameters

MULTIPLY *dataname1 datamane2*

dataname1, *dataname2*= (C) Names (including path) of the two data arrays that are multiplied.. Note that this opertation multiplies each individual array member of *dataname1* by the equivalent array member in *dataname2* and stores the results in the same array location in *dataname2*. Therefore, both arrays must be the same length. This command changes the values contained in the *dataname2* but *dataname1* remains unchanged. No default.

Note: Example: The following example redefines each member of array dat2 as follows:
dat2(*i*) = dat1(*i*) * dat2(*i*)

```
DATA
    MULTIPLY dat1 dat2
    END
```

OPEN Subcommand
DATA Input Group **DATA-OPEN**

Purpose: To open a data array within the Data Manager.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM input parameters

OPEN *dataname ndimension ndimi ndimj ndimk datatype*

dataname = (C) Name of the Data Manager array to open. This array name must not already exist.
No default.

ndimension = (I) Number of dimensions of the data array.
Form 1 to 3.
No default.

ndimi = (I) Range of the array's i-index
Default=0.

ndimj = (I) Range of the array's j-index
Default=0.

ndimk = (I) Range of the array's k-index
Default=0.

datatype = (C) The type of data contained in this data array.
Any of: I = integer numbers
F = floating point real numbers
C = character*4 character data
No default.

OUT Subcommand
DATA Input Group

DATA-OUT

Purpose: To save model information and snapshots of field variable arrays to the Data Output File for future postprocessing/plotting using the **Review** program. To export data for later use by the **PZFlex** software or for other array manipulation prposes, use the **OUT1** command instead.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: None.

PCOM SCOM _____ input parameters _____

OUT *dataname*

dataname = (C) Data Manager array name or MODL.
If MODL is input, then the complete model geometry is exported to the flxdato file. To plot snapshot results using **Review**, export the model prior to performing any other **DATA OUT** functions.
No default.

Notes: 1. See the section *Files and Formats* for a description of the format of the Data Output File.
2. If the **OUT** subcommand is used to write the first data array to a Data Output File, any succeeding writes to the same file must also be done with the **OUT** subcommand. If it is desired to use the **OUT1** subcommand at a later time, the **FILE** subcommand must be used to define a new Data Output File in order to allow the use of the **OUT1** subcommand.

Example:

```
DATA
    OUT MODL /* export all model geometry data to the Data Output File
    OUT PRES /* export the current pressure field to the Data Output File
    (etc.)
```

OUT0 Subcommand
DATA Input Group

Purpose: To write the values contained within a Data Manager array to the Data Output File using the old **FLEX1H DATA OUT** command format. This option is provided only for backwards compatibility. In general, users should use the **OUT** or **OUT1** subcommands instead.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM	SCOM	input parameters
	OUT0	*dataname*

dataname = (C) The Data Manager array name. No default.

Notes: 1. See the section *Files and Formats* for a description of the format of the Data Output File created by the **OUT0** subcommand.

2. If the **OUT0** subcommand is used to write the first data array to a Data Output File, any succeeding writes to the same file must also be done with the **OUT0** subcommand. If it is desired to use the **OUT** subcommand at a later time, the **FILE** subcommand must be used to define a new Data Output File in order to allow the use of the **OUT** subcommand.

OUT1 Subcommand
DATA Input Group

DATA-OUT1

Purpose: To write the values contained within a Data Manager array to the Data Output File. This option should be used if the user wishes to export data for later use by the **PZFlex** software or other software programs. It provides the most flexibility for data array import/export. To save snapshots of field variable arrays to be read into the **Review** program for plotting, use the **DATA OUT** subcommand instead.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM	SCOM	input parameters
	OUT1	*dataname ibegin iend jbegin jend kbegin kend*

dataname = (C) Data Manager array name.
No default.

ibegin, iend = (I) Beginning and ending I-indices of the portion of the array to be exported.
Default: *ibegin*=1, *iend*=maximum I

jbegin, jend = (I) Beginning and ending J-indices of the portion of the array to be exported.
Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of the portion of the array to be exported.
Default: *kbegin*=1, *kend*=maximum K

Notes: 1. See the section "Files and Formats" for a description of the format of the Data Output File created by the **OUT1** subcommand.
2. If the **OUT1** subcommand is used to write the first data array to a Data Output File, any succeeding writes to the same file must also be done with the **OUT1** subcommand. If it is desired to use the **OUT** subcommand at a later time, the **FILE** subcommand must be used to define a new Data Output File in order to allow the use of the **OUT** subcommand.

PHAD (phasor add) Subcommand
DATA Input Group **DATA-PHAD**

Purpose: To add two phasor arrays.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: Must follow the point at which the data array is created.

PCOM SCOM input parameters

PHAD *ampfrom phasfrom scale ampto phasto ibegin iend jbegin jend kbegin kend*

ampfrom, phasfrom= (C) Data Manager arrays containing the values to be scaled and added. The arrays are not changed by this subcommand. No default.

scale= (F) Scale factor to scale *ampfrom* before adding them to *ampto*. Default = 0.

ampto, phasto= (C) Data Manager arrays to add values to. The arrays are changed by this subcommand. No default.

ibegin, iend= (I) Beginning and ending I-indicies of data array locations to be modified by this subcommand.
Default: *ibegin*=1, *iend* = maximum I of *ampto*

jbegin, jend= (I) Beginning and ending J-indicies of data array locations to be modified by this subcommand.
Default: *jbegin*=1, *jend* = maximum J of *ampto*

kbegin, kend= (I) Beginning and ending K-indicies of data array locations to be modified by this subcommand.
Default: *kbegin*=1, *kend* = maximum K of *ampto*

Note: To combine data arrays with the **PHAD** subcommand without actually changing the values stored in *ampto*, a working copy of the array may be created using the **COPY** subcommand which can then be used by the **PHAD** subcommand for the desired combination.

POLR (polar conversion) Subcommand
DATA Input Group **DATA-POLR**

Purpose: To convert between (real,imaginary) and (amp,phase).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **DATA** Group: Must follow the point at which the data arrays are created.

PCOM SCOM input parameters

POLR *real imag option amp phas ibegin iend jbegin jend kbegin kend*

real, imag = (C) Data Manager arrays containing the (real, imaginary) values. No default.

option= (C) Option: *to*: real, imag >> amp,phas
from: real,imag << amp,phas
Default: *to*.

amp, phas= (C) Data Manager arrays containing the (amp,phas) values. No default.

ibegin, iend= (I) Beginning and ending I-indicies of data array locations modified by this subcommand.
Default: *ibegin*=1, *iend* = maximum I of *phas*

jbegin, jend= (I) Beginning and ending J-indicies of data array locations to be modified by this subcommand.
Default: *jbegin*=1, *jend* = maximum J of *phas*

kbegin, kend= (I) Beginning and ending K-indicies of data array locations to be modified by this subcommand.
Default: *kbegin*=1, *kend* = maximum K of *phas*

Notes: 1. The POLR conversion can be done in place if *amp* and *real* are the same array name and *phas* and *imag* are the same array name.
2. *phas* is in Radians. *real = amp*cos(phas); imag = amp*sin(phas)*

REMAP Subcommand
DATA Input Group **DATA-REMAP**

Purpose: To interpolate elemental or nodal data from a source grid into a destination grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM input parameters

REMAP *pathname source_array destination_array axis nsample &*
ibegin iend jbegin jend kbegin kend

pathname = (C) Full path and name of an existing disk file (flxdato file) containing the source grid. This file must have been written with a DATA OUT command and contain coordinate and geometry information in addition to the source data array that will be remapped.
No default.

source_array = (C) Name of the elemental or nodal data array in the source grid whose values will be remapped.
No default.

destination_array = (C) Name of the existing elemental or nodal data manager array in the destination grid (current model) that *source_array* will be mapped into.
Default = *source_array*.

axis = (C) Name of a predefined local coordinate system defined using the AXIS command to describe the location and orientation of the source grid within the destination grid.
Default is standard global coordinate system (ie. no transformation between source and destination grids).

nsample = (I) Number of sampling points per destination element in each coordinate direction. Only used for element data. See Note 1.
Default = 1.

ibegin, iend = (I) Beginning and ending I-indices of destination grid data array locations to limit mapping.
Default: *ibegin* =1, *iend* = maximum I of destination grid

jbegin, jend = (I) Beginning and ending J-indices of destination grid data array locations to limit mapping.
Default: *jbegin* =1, *jend* =maximum J of destination grid

kbegin, kend = (I) Beginning and ending K-indices of destination grid data array locations to limit mapping.
Default: *kbegin*=1, *kend*=maximum K of destination grid

Notes: 1. *nsample* can be increased to achieve better resolution for the case where the source grid has a finer mesh than the destination grid. If *nsample* is greater than 1 then the values at the sampling points are averaged to obtain the mapped value for the destination element. Increasing the number of sampling points can slow down the processing time. Processing time will increase by *nsample* cubed for 3D meshes.

2. Data values in *destination_array* are overwritten by the values of *source_array* only if a *source_array* element actually maps onto the *destination_array* element, otherwise the *destination_array* element retains its original value.

SCAL (Ssale) Subcommand
DATA Input Group **DATA-SCAL**

Purpose: To allow a data group stored in the Data Manager to be scaled by a scale factor. The data values within the data group are permanently changed with this command.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **DATA** Group: None.

PCOM SCOM input parameters

SCAL *dataname scalevalue ibegin iend jbegin jend kbegin kend*

dataname = C) The Data Manager array name.
No default.

scalevalue = (F) Scale factor.
No default.

ibegin, iend = (I) Beginning and ending I-indices of the section of the array to be scaled.
Default: *ibegin*=1, *iend*=maximum I

jbegin, jend = (I) Beginning and ending J-indices of the section of the array to be scaled.
Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of the section of the array to be scaled.
Default: *kbegin*=1, *kend*=maximum K

Note: This option is typically used to change the units of a data group for display or presentation in a postprocessing step. For example, to change the units of the SGXX stress data group from ksi to psi, the following command is used.

Example 1:
```
DATA
   SCAL SGXX 1000.
   END
```

TABL Subcommand
DATA Input Group **DATA-TABL**

Purpose: To create and assign values to a 2 dimensional Data Manager array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM input parameters

TABL *dataname idim jdim*
data(1,j) j=1,jdim
data(2,j) j=1,jdim
....

dataname = (C) The data manager array name that will contain the data once it has been imported from the Data Input File. No default.

idim = (I) The number of rows.

jdim = (I) Number of columns.

data(i,j) = (F) The data values

VPHASE (Phase Velocity) Subcommand
DATA Input Group **DATA-VPHASE**

Purpose: To fill an array with phase velocity vs. frequency values derived from an array containing wave number vs. frequency data.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **DATA** Group: None.

PCOM SCOM input parameters

VPHASE *datanamefrom datanameto*

datanamefrom = (C) The Data Manager array containing wave number vs. frequency data. The array must be data manager type 7 (mapped to coordinate vectors). This array will not be modified by this subcommand.
No default.

datanameto = (C) The Data Manager array to be filled with phase velocity data. A data manager type 7 array must be previously set-up to accept the data. See note 1.
No default.

Notes: 1. The complete command sequence to create a phase velocity array from an existing wave number array is for example:

```
DATA OPEN phasevel 2 $idim $jdim * f
     MGR STRUCTURE phasevel freq frequency $frangmin $frangmax &
                                pvel phase_velocity $vrangmin $vrangmax
     VPHASE wavearray phasevel
     END
```

DRLX (dynamic relaxation) Input Command

Purpose: To define a dynamic relaxation process to cause the response to damp to the static solution.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM _____ input parameters

DRLX *relaxfreq nstavg stoptol*

or

DRLX AUTO *nsample frqfac nstavg stoptol*

or

DRLX AUTO2 *relaxfreq relaxratio nstavg stoptol*

relaxfreq = (F) Frequency of response to be critically damped in radians/unit time.
Default = 0.0, indicates no damping.

relaxratio= (F) Target ratio of
(energy dissipation rate / |rate of change of kinetic energy|)
Default = 0.5.

nsample= (I) Algorithm requires the rate of change of internal forces. This is approximated by a difference over *nsample* model timesteps.
Default = Largest subcycling ratio.

frqfac= (F) Ratio of critically damped frequency to that computed by the Rayleigh quotient. Default = 1.0.

nstavg= (F) An exponentially decaying running average of a velocity norm over *nstavg* timesteps is computed to assess convergence. In approximate terms, the velocity norm must stay below the specified tolerance for *nstavg* timesteps for convergence. Default = 100.

stoptol= (F) Relative tolerance used by the STAT option on the EXEC sommand. When the running average velocity is less than *stoptol* * its peak prior value, execution is terminated. Default = 1.e-4.

DRLX

Notes: 1. Different frequencies may be damped at different times during an analysis by redefining the value of *relaxfreq.*
2. In general, the loading of a nonlinear model when performing a dynamic relaxation analysis should be applied using a ramp in time rather than as a step function.
3. Response modes of the model with frequencies less than *relaxfreq* are over critically damped, while those with frequencies above *relaxfreq* are under critically damped.
4. The AUTO option adaptively changes the relaxation frequency using a Rayleigh quotient.
5. The AUTO2 option adaptively changes the relaxation constant from its initial value to maintain the requested *relaxratio.* It is used primarily for nonlinear models where an appropriate *relaxfreq* cannot be chosen *apriori.* The idea was proposed by (Cundall, 1982).
6. The DRLX array contains time-varying quantities of interest. In all cases, DRLX(7) is a mass-weighted velocity norm. DRLX(8) is the peak prior value of DRLX(7), DRLX(9) is a moving average of DRLX(7), DRLX(10) is the ratio: DRLX(9)/DRLX(8), and DRLX(11) is the time at which DRLX(7) reached peak. For the AUTO option, DRLX(1) contains the critically damped angular frequency.

DRLX (dynamic relaxation) Input Command

DRLX
(extension for 2D plane strain models)

Purpose: In addition to the standard DRLX command described above, the following extension of the command for 2D plane strain models provides control over mass and stiffness proportional damping parameters. The default values of damping are zero.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM _____ input parameters

DRLX *typedamp1 dampvalu1 typedamp2 dampvalu2* etc.

typedampn = (C) Type of damping parameter being specified.
Any of:
- MASS - mass proportional damping
- STIF - stiffness proportional damping for standard continuum element modes
- HGLS - stiffness proportional damping for continuum element hourglass modes

dampvalun= (F) Damping coefficient to use for the preceding damping type specified.

Note:
1. All three types of damping or any subset may be specified on a single command line. The value of a damping coefficient remains unchanged unless explicitly modified with the DRLX command.
2. The damping in the model may be changed at any time during a calculation.
3. For a particular response mode of the system, the fraction of critical damping, c_i , is defined by:

$$c_i = \frac{1}{2}\left(\frac{\alpha}{\omega_i} + \beta\omega_i\right)$$

where α and β are the mass and stiffness proportional damping coefficients specified on the **DRLX** command, and ω is the frequency of the response mode in radians per unit time.

ECHO Input Command

ECHO

Purpose: First step in computing the "round-trip" response of a transducer. This command is issued during the "send" phase to produce boundary input for a subsequent "receive" calculation for the same model. It assumes an infinite reflector at a specified position far from the transducer. The reflector is assumed to be oriented parallel to the transducer face.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow **GRID** command and precede **PRCS** command.

PCOM SCOM _____ input parameters

ECHO

NODE	*ibegin iend jbegin jend kbegin kend iinc jinc kinc*
RCRD	*xcrdrefl ycrdrefl zcrdrefl option*
PULS	*pulse_duration quad_order stor_opt nsection zlength*
SYMM	*symm_condition*
END	

Notes:
1. As the return signal is assumed to be a normally incident plane wave, the reflector must be far enough away and properly oriented to make this assumption valid.
2. Time -omain Kirchhoff extrapolation is used to extrapolate through a homogeneous nonabsorbing fluid filling the space between the surface defined by the **NODE** command(s), and the reflector.
3. Summary information about this option is written to the flxprt file during the **PRCS** step.
4. The current value of extrapolated pressure is stored in a single value array named ECHO. The **POUT** or **OUTP** command group can be used to save a time history of thie extrapolated pressure as shown here:
   POUT
   HIST ECHO
   END
5. The time delay relative to model time is printed to the flxprt file at the end of the job.

NODE Subcommand
ECHO Input Group

ECHO-NODE

Purpose: To define a surface for use in extrapolating the send pulse out to the reflector and back.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **ECHO** Group: None.

PCOM SCOM input parameters

NODE *ibegin iend jbegin jend kbegin kend iinc jinc kinc*

ibegin, iend= (I) Beginning and ending I-indicies of nodes on this portion of the echo surface.
Default: *ibegin* = 1, *iend* = maximum I.

jbegin, jend= (I) Beginning and ending J-indicies of nodes on this portion of the echo surface.
Default: *jbegin* =1, *jend* = maximum J.

.

kbegin, kend= (I) Beginning and ending K-indicies of nodes on this portion of the echo surface.
Default: *kbegin* = 1, *kend* = maximum K.

iinc, jinc, kinc= (I) Increment in nodes to more coarsely sample the surface.
Default: *iinc* = *jinc* = *kinc* = 1.

Notes: 1. The node command is used to define a line of nodes in 2D or a 2D surface of nodes in 3D. The surface must lie entirely within the homogeneous, nonabsorbing fluid through which the pulse is to propagate.

2. All of the elements on the surface must be processed at the same time step. If they are not, the code issues an error message and exits.

RCRD (reflector coordinates) Subcommand
ECHO Input Group **ECHO-RCRD**

Purpose: To define the reflector position in model coordinates.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence
within **ECHO** Group: None.

PCOM SCOM _____ input parameters

RCRD *xcrdrefl ycrdrefl zcrdrefl option*

xcrdrefl = (F) x-coordinate of the point on the reflector directly above the center of the transducer. No default.

ycrdrefl = (F) y-coordinate of the point on the reflector directly above the center of the transducer. No default.

zcrdrefl = (F) z-coordinate of the point on the reflector directly above the center of the transducer. No default.

option= (C) Either REFL or OBSR. For REFL, these are reflector coordinates, and a boundary file is written for the receive problem. For OBSR, these are observation point coordinates, and no boundary file is written.
Default = REFL.

Notes: 1. If a full model of the transducer is used, the reflection point should be directly above the center of the model. If a half or quarter model is used, the reflection point should be directly above the edge or corner of the model. For axisymmetric models, the reflection point must be on the axis.
2. The OBSR option permits extrapolation to a single far-field point. For half or quarter models, this point must lie on the symmetry plane(s). For axisymmetric, the point is not required to be on the axis, but it should be over the center of the model.

PULS (pulse) Subcommand
ECHO Input Group **ECHO-PULS**

Purpose: To define the pulse duration.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence
within **ECHO** Group: None.

PCOM SCOM _____ input parameters

PULS *pulse_duration quad_order stor_opt nsection zlength*

pulse_duration = (F) Pulse duration. The length of the pulse to be computed for the receive calculation. This must be known before the calculation starts. A reasonable upper bound should be input. Default = 0.

quad_order= (I) Gaussian quadrature order. Number of integration points in each direction per element Default = 1.

stor_opt= (C) Option to store or calculate Green's functions for 2D, 2.5D or axisymmetric models. Either STOR or CALC. CP time can be dramaticaly reduced by storing them, but the required memory can be large. See notes below. Default = STOR.

nsection= (I) Number of elements in the theta or z directions for axisymmetric or 2.5D extrapolations respectively. Default = 36 for axisymmetric; computed by code and printed to flxprt file for 2.5D. See note 3 below.

zlength= (F) Length in z-direction for 2.5D extrapolation. Enter 0., * or blank for 2D (infinite zlength).

Notes:
1. 2.5D means approximating 3D data by uniform 2D data of length *zlength*.
2. Additional words of memory required by the STOR option on the PULS subcommand is roughly (*9*nelem*nsection*) for 2.5D or axisymmetric models and (*2*nsteps*nelem*) for 2D. Here *nelem* is the number of elements on the surfaces defined by NODE subcommands, and *nsteps* is the number of model timesteps required to run *Pulse_duration.*
3. The number of sections for 2.5D or axisymmetric should be chosen so that the travel times from adjacent sections to the reflector or observation point are approximately equal to the timestep. This requires a section size about the same as a finite element in the near field but much larger sections can be used for far field reflectors.

SYMM (symmetry) Subcommand
ECHO Input Group

ECHO-SYMM

Purpose: To define symmetries for the echo calculation.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **ECHO** Group: None.

PCOM SCOM input parameters

SYMM *symmetry_condition*

symmetry_condition= (C) Any of FULL, HALF, or QRTR. Input FULL if a finite element model of the entire transducer is used, HALF for a half model, and QRTR for a quarter model. Default = FULL.

EXEC (execute) Input Command

Purpose: To compute the response of the model.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence:Must follow the **PRCS** command.

PCOM _____ input parameters

EXEC *ntime endtime (static_option) print_option*

or

EXEC CYCL *ncycle print_option*

or

EXEC RINGDOWN (array) *ntimering nstavg stoptol ibeg iend jbeg jend kbeg kend offset*

or

EXEC RINGDOWN electrode *eoptn ename ntimering nstavg stoptol offset*

ntime = (I) Number of problem time steps to execute.
Default = 1

endtime = (F) Specifies the maximum problem time at which the this **EXEC** command will conclude. If *endtime* is reached before *ntime* timesteps are executed the **EXEC** command is concluded. Likewise, if *ntime* timesteps are executed before *endtime* is reached, the **EXEC** command is concluded. If 0.0 or default, this parameter is ignored.

ncycle= (I) The number of timesteps equivalent to *ncycle* complete cycles of the driving frequency are executed. See note 4 below. Default = 1.

static_option= (C) Static solution option. Enter STAT for static solution. STAT will terminate execution at the earliest of *ntime* timesteps, time = *endtime* or convergence of the dynamic relaxation procedure. The DRLX command must be input prior to EXEC with the STAT option.

print_option= (C) Print option. Any of: ON or OFF. If set to ON, normal printout of execution statistics is produced. If OFF, no printout is produced. Default = ON

ntimering = (I) Number of problem time steps to execute ringdown evaluating option. See Note 5. Default = 100,000

PCOM input parameters

nstavg = (I) Number of problem time steps over which to average the nodal velocities. Default = 100

stoptol = (F) Relative tolerance. When the running average velocity becomes less than *stoptol* * its peak prior value, execution is terminated. Default = 1.e-3.

ibeg, iend = (I) The beginning and ending I-indices of nodes that bound the ringdown controlling region .
Default: *ibeg* = 1, *iend* = maximum I

jbeg, jend = (I) The beginning and ending J-indices of nodes that bound the ringdown controlling region .
Default: *jbeg* = 1, *jend* = maximum J

kbeg, kend = (I) The beginning and ending K-indices of nodes that bound the ringdown controlling region .
Default: *kbeg* = 1, *kend* = maximum K

electrode = (C) This is a keyword. Enter electrode

eoptn = (C) Any of v, q or i. v=voltage, q=charge, i=current. Default = q.

ename = (C) Electrode namet. Default = first electrode.

offset = (C) Enter the keyword OFFSET to use the velocity (or other array) norm at the time the EXEC command is issued as an offset for computing the peak and average ringdown quantities. Useful in the case of static bias, for example. Default is no offset used.

Notes: 1. If the **PRCS** command has not been previously input, **EXEC** automatically performs the **PRCS** command.
2. Any number of **EXEC** commands may be used within one or in subsequent restart runs.
3. The *endtime* option is useful for large deformation runs in which the model time step may change and it is not possible to know exactly how many time steps are required to reach a desired time.
4. For the CYCL option, the driving frequency must be defined by a **FUNC SINE** command. The time span of the requested number of cycles is approximated by the nearest (integer) number of timesteps.
5. For the RINGDOWN option, timesteps are executed until the average normalized nodal velocity is less than *stoptol* or *ntime* timesteps have been executed, whichever occurs first.

EXTR (extrapolation) Input Command

Purpose: To create an output data file (flxext) for use in various types of extrapolation using Review. This includes the generation of beam pattern plots and Kirchhoff time-domain extrapolation. The file contains time histories of grid pressure for user-defined data planes in the model. This option is available for 2D plane strain, 2D axisymmetric and 3D models.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow **GRID** command and precede the **PRCS** command.

PCOM	SCOM	input parameters
EXTR		
	REF	*refoption xref yref zref*
	DEFN	*extoption*
	DRIV	*driv_array*
	NODE	*ibegin iend jbegin jend kbegin kend iinc jinc kinc*
	TIME	*timebegin irate*
	END	

REF (reference point) Subcommand
EXTR (extrapolation) Input Group **EXTR-REF**

Purpose: To specify a reference point that defines the interior and exterior sides of an extrapolation surface.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence within **EXTR** Group: None.

PCOM	SCOM	input parameters
	REF	*refoption xref yref zref*

refoption = (C) Either IN or OUT. If set to IN, a line drawn from the reference point to the extrapolation surface touches the inside of the surface. If set to OUT, the line touches the outside of the surface. The extrapolation process extrapolates the solution outside the surface.

xref, yref, zref = (F) x- , y-, and z-coordinates of the reference point. Default: *xref* = *0.0*, *yref* = *0.0*, *zref* = *0.0*

DEFN (define) Subcommand
EXTR (extrapolation) Input Group **EXTR-DEFN**

Purpose: To define the type of the extrapolation surface.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence within **EXTR** Group: Must precede the **NODE** subcommand that defines the location of the extrapolation surface.

PCOM	SCOM	input parameters
	DEFN	*extoption*

extoption = (C) Set to either FOUR or KIRC. If set to FOUR, basic information required for Fourier extrapolation (the option used for beam pattern plots) is saved on the extrapolation surface. If set to KIRC, the additional information required for Kirchhoff extrapolation is also saved. The KIRC option saves twice as much information as the FOUR option.

DRIV Subcommand
EXTR (Extrapolation) Input Group **EXTR-DRIV**

Purpose: To add drive function information to the extrapolation file as a convenience. This can be used by FlexLab to compute TVR.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
Within **EXTR** Group: None.

PCOM SCOM input parameters

DRIV *driv_array ibeg iend jbeg jend kbeg kend*

driv_array = (C) The name of the drive function to be saved to the flxext file. No default — typically FUNC would be specified.

ibeg…kend = (C) The indicies of drive array values to be saved to the flxext file at each timestep.
Default = 1. Typically, a single value at each timestep would be specified.

NODE subcommand
EXTR (extrapolation) input group **EXTR-NODE**

Purpose: To specify the nodes that make up extrapolation surface. Multiple "node" commands may be input when defining Kirchhoff data planes, only one "node" command, however, is allowed when defining a Fourier data plane

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
Within **EXTR** Group: Must follow the "defn" subcommand that is associated with these nodes.

PCOM SCOM input parameters

NODE *ibegin iend jbegin jend kbegin kend iinc jinc kinc*

ibegin, iend = (I) Beginning and ending I-indices of nodes on the extrapolation surface.
Default: ibegin = 1, iend = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes on the extrapolation surface.
Default: jbegin = 1, jend = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes on the extrapolation surface.
Default: kbegin = 1, kend = maximum K

iinc, jinc, kinc = (I) Increments in i,j,k. Default = 1. See note 4.

Notes: 1. The "node" subcommand must define a plane of nodes for a 3D model. For example, if ibegin = iend , then jbegin = jend and kbegin = kend.
2. All surfaces defined with the "node" subcommand must lie entirely within the homogeneous medium (such as vacuum, water, or air) through which the extrapolation is to be performed. Material damping is not included in the extrapolation.
3. All computational zones that the extrapolation surface passes through must have the same time step.
4. Typically, the increments should be set to 1 for accurate extrapolation. Larger values should be used only when the model is very highly discretized. At least 20 points per wavelength should be retained for the extrapolation.

TIME Subcommand
EXTR (extrapolation) Input Group **EXTR-TIME**

Purpose: To reduce the amount of data written to the extrapolation data file by specifying the beginning time at which data on the extrapolation suface should be saved. If not input, data is saved from the beginning of the calculation. This option can be useful if the first wave arrivals at the extrapolation surface occur at a time significantly greater than 0.0.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **EXTR** Group: None.

PCOM SCOM input parameters

TIME *timebegin*

timebegin = (F) Beginning time at which extrapolation data will start being saved to the extrapolation file.
Default = 0.0

irate = (I) Output rate. Data will be written to the extrapolation file each irate timesteps.
Default = 1

FFLD (freefield) Input Command

FFLD

Purpose: To define information about freefield boundary files. This command is used to define a freefield boundary file for use as a boundary condition for the model or to request that a freefield boundary file be generated during the computation for future use.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow **GRID** command and precede the **PRCS** command.

PCOM SCOM input parameters

FFLD

Commands for importing freefield input files

FILE	*filename*
IN	*axisname timeshift fuzzfactor option nocheck*

Commands for generating freefield output files

OUT	*ibeginvel iendvel jbeginvel jendvel kbeginvel kendvel* & *ibeginstr iendstr jbeginstr jendstr kbeginstr kendstr*
RATE	*irate_out*
CORD	*active_coord(s)*

END

Notes:
1. A freefield boundary file contains the information to apply freefield boundary conditions on the sides of a computational model. This boundary condition provides a way to prescribe freefield velocities at the boundary, detect scattered waves that result due to the presence of an object within the model, and absorb these scattered waves by adjusting the prescribed velocities appropriately.
2. A freefield boundary file can become very large if it covers a large spatial distance, spans a long time duration, or has to store all components of stress and velocity.
3. The freefield file is restricted to standard partition type data where only one coordinate varies in each index direction. The model accessing the freefield data is not restricted in this manner.

FILE Subcommand
FFLD (freefield) Input Group **FFLD-FILE**

Purpose: To identify an existing freefield boundary file that contains the information for FFLD boundary conditions in this calculation.

Use Is: Required if FFLD boundary conditions have been specified.

Multiple Use: Not allowed.

Order Dependence
within **FFLD** Group: None.

PCOM	SCOM	input parameters
	FILE	*filename*

filename = (C) Name of the freefield boundary file. Typically this file has a name containing the identifier "flxffl."

Notes: 1. A freefield boundary file specified with the **FILE** subcommand is, by default, assumed to:

(a) have been created in the global coordinate system of the present model

(b) have time information consistent with the present model

(c) have spatial data that properly encompasses the boundaries to which it is applied

(d) apply motions corrected for scattered waves within the model. This is equivalent to boundary option = STND

2. The **IN** subcommand can be used to change these defaults if desired.

IN Subcommand
FFLD (Freefield) Input Group **FFLD-IN**

Purpose: To change default characteristics of a freefield boundary file defined by the **FILE** subcommand.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
Within **FFLD** Group: None.

PCOM SCOM input parameters

IN *axisname timeshift fuzzfactor option [symmetry options] nocheck*

axisname = (C) Name of the axis definition which defines the local freefield coordinate system. Must have been previously defined by the AXIS command group or set to STND if the standard global coordinate system of the current model is to be assumed. Defaults=STND.

timeshift = (F) The value to add to the freefield file time data to align it with the time of the current model.
Default = 0.0

fuzzfactor = (F) The parameter which defines the tolerance on freefield boundary coordinates. If numerical precision issues cause the code to determine that boundary nodes fall outside the spatial data contained on the freefield boundary file, increasing the fuzzfactor will increase the allowed coordinate mismatch before causing an error.
Default = .001

option = (C) Boundary condition option. Either: STND or VEL
If set to STND: motions are applied and scattered waves are absorbed by the boundary.
If set to VEL: motions are applied and no absorption of scattered waves occurs. This is a pure kinematic boundary. Default = STND

symmetry options = (C) Enter up to two of −X, +X, -Y, +Y, -Z, +Z to indicate symmetry planes on the freefield file. For example, -X causes the freefield data to be mirrored about the minimum X coordinate in the file when it is mapped to the current model. Leave blank if there are no symmetry planes.

(**IN** subcommand continues on next page)

nocheck= (C) End-of-file check option. Enter NCHK to allow execution beyond the time range on the freefield file, or CHEK to terminate at end of file. Freefield velocities and stresses will be assumed to be zero for times beyond the end. Default: NCHK. In either case, the calculation may be restarted and execution will continue with zero freefield stressses and velocities.

OUT Subcommand
FFLD (freefield) Input Group **FFLD-OUT**

Purpose: To define the region of the current freefield model for which freefield boundary information will be saved to a file.

Use Is: Optional. Used only if generating a freefield boundary file.

Multiple Use: Allowed.

Order Dependence
within **FFLD** Group: None.

PCOM SCOM input parameters

OUT *ibeginvel iendvel jbeginvel jendvel kbeginvel kendvel* &
ibeginstr iendstr jbeginstr jendstr kbeginstr kendstr

ibeginvel, iendvel = (I) Beginning and ending I-indices of nodes bounding the spatial region to save velocity data for.
Default: *ibeginvel* = 1, *iendvel* = maximum I

jbeginvel, jendvel = (I) Beginning and ending J-indices of nodes bounding the spatial region to save velocity data for.
Default: *jbeginvel* = 1, *jendvel* = maximum J

kbeginvel, kendvel = (I) Beginning and ending K-indices of nodes bounding the spatial region to save velocity data for.
Default: *kbeginvel* = 1, *kendvel* = maximum K

Input the following information only if different from the velocity ijk window above.

ibeginstr, iendstr = (I) Beginning and ending I-indices of nodes bounding the spatial region to save stress data for.
Default: *ibeginstr* = 1, *iendstr* = maximum I

jbeginstr jendstr = (I) Beginning and ending J-indices of nodes bounding the spatial region to save stress data for.
Default: *jbeginstr* = 1, *jendstr* = maximum J

kbeginstr kendstr = (I) Beginning and ending K-indices of nodes bounding the spatial region to save stress data for.
Default: *kbeginstr* = 1, *kendstr* = maximum K

FFLD-OUT

Notes: 1. The freefield boundary file produced is a flxffl file.
2. The flxffl file produced can be very large, as it contains the velocities and stresses for all the nodes and elements contained in the requested ijk regions for each time step saved. The **RATE** and **CORD** subcommands can be used to reduce the amount of data saved to the file.
3. For axisymmetric calculations, flxffl files are written in cylindrical coordinates (see the AXIS command) where z is the axial direction, and r is y for axix or x for axiy.

RATE Subcommand
FFLD (freefield) Input Group **FFLD-RATE**

Purpose: To specify the time step sampling rate at which freefield data is written to the freefield boundary file.

Use Is: Optional. If not specified, the sampling rate defaults to 1.

Multiple Use: Not allowed.

Order Dependence
within **FFLD** Group: None.

PCOM SCOM input parameters

RATE *irate_out*

irate_out = (I) Sampling rate at which data is written to the freefield boundary file.
Default = 1

CORD Subcommand
FFLD (freefield) Input Group **FFLD-CORD**

Purpose: To specify a reduced set of coordinate axes in order to reduce the amount of data saved to the freefield boundary file being generated. Use this command if using a 2D model to represent a freefield model that is actually 1D.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **FFLD** Group: None.

PCOM	SCOM	input parameters
CORD		*coordinate(s)*

coordinate = (C) Valid coordinate axis along which field values such as velocity and stress vary. Any of X, Y or Z. Input as many as actually apply. Default is all coordinate axes applicable to the current model.
For 2D models, this is X and Y.
For 3D models this is X, y, and z.

FLOW Input Command

Purpose: To implement one way flow convection.

Use Is: Optional.

Multiple Use: Not Allowed.

Order Dependence:Must precede PRCS command

PCOM SCOM input parameters

FLOW

MATR *mat1 mat2 mat3...matn*

Notes: 1. Implemented in small deformation, standard partition only.
2. Currently only linear fluids are considered
3. The algorithm is "1-way" in that velocities and pressures are convected by the prescribed flow velocity, but the flow velocities are unaffected by the computed solution.
4. Flow velocities are user prescribed (e.g. using a SET command) in the nodal arrays xvflow(indgrd, jndgrd,kndgrd), yvflow(indgrd,jndgrd,kndgrd), zvflow(indgrd,jndgrd,kndgrd).

MATR Subcommand
FLOW Input Group

Purpose: To specify materials for which flow velocities apply. If input, Flow velocities are set to zero for all nodes in contact with an unspecified material.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **FFLD** Group: None.

PCOM	SCOM	input parameters
	MATR	*mat1 mat2 mat3…matn*

mat1,…matn = (C) List of materials for which specified FLOW velocities are active. No defaults.

FUNC (function) Input Command

Purpose: To define the characteristics of the special purpose function.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must precede any command that utilizes the function.

PCOM SCOM input parameters

FUNC

NAME	*namfnc*
BLAK	*centerfrequency amplitude tdelay*
CHIRP	*amplitude timeshift frq1 frq2 duration rampup rampdown*
GAUS	*width amplitude tdelay*
HIST	*histname scal adopt tdelay*
SINE	*frequency amplitude phaseshift nperiod valuadd rampcycle tdelay*
STEP	*amplitude2 steptime amplitude1 tdelay*
TDELAY	*option*
WVLT	*frequency amplitude gamma nu tdelay*

END

Notes:
1. Only one of the function types may be specified in a single job.
2. Once defined, the time history represented by the function defined by the **FUNC** command may be used in any command that requires inputting a time history name by specifying FUNC instead of a time history name.
3. If the **FUNC** option is used in a computation, a time history of the function can be requested using either the **POUT** or **OUTP HIST** option as follows:

POUT
HIST FUNC

NAME Subcommand
FUNC (Function) Input Group **FUNC-NAME**

Purpose: To set the current function name.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **FUNC** Group: None. All following FUNC subcommands will apply to the currently named function until another NAME command is encountered.

PCOM SCOM input parameters

NAME *namfnc*

namfnc = (C) Unique name for this function.
Default = blank.

ASTEP (Apodized Step)Subcommand
FUNC (Function) Input Group **FUNC-ASTEP**

Purpose: To define function parameters for an apodized step function.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
Within **FUNC** Group: None.

PCOM SCOM input parameters

ASTEP *amplitude2 steptime amplitude1 tdelay type exponent band & amplitude3 steptime2*

amplitude2 = (F) Amplitude of step function for *steptime* < time < *steptime2*. Default = 0.0

steptime = (F) Time at which first step occurs. Default = *band* / 2.

amplitude1 = (F) Amplitude of step function for time ≤ *steptime*. Default = 0.0

tdelay = (F) Time delay. Default = 0.0. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.

type = (C) Apodization function type. Either of: LINEAR or COS. Default= COS.

exponent = (F) Exponent for cosine function. Default = 1.0.

band = (F) Apodization band. Default = 0.0.

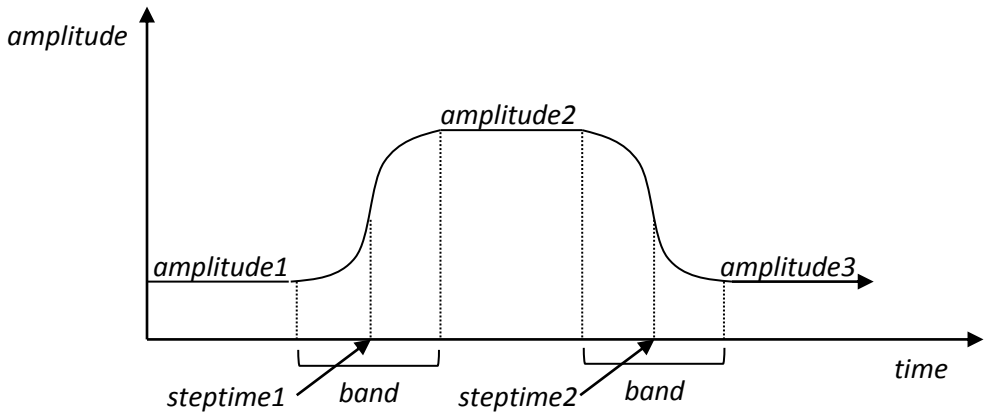
amplitude3 = (F) Amplitude of step function for *steptime2* < time Default = 0.0

steptime2 = (F) Time at which second step occurs. Default = no 2nd step.

(description of **ASTEP** subcommand continues on next page)

ASTEP (Apodized Step) Subcommand
FUNC (Function) Input Group

FUNC-ASTEP
(continued)



Notes: 1. Non-zero values of *tdelay* shift all time inputs by *tdelay*.

2. cosine function is given by:

$value = A1 + (A2-A1) * [\,(1-\cos(theta))/2.\,]^{exponent}$

BLAK (Blackman Harris) Subcommand
FUNC (function) Input Group **FUNC-BLAK**

Purpose: To specify parameters defining the functional form of the 2nd derivative of the Blackman Harris window. This function produces a wavelet in time whose frequency content is a Gaussian distribution centered at the center frequency.

Use Is: Optional.

Multiple Use: Not allowed.

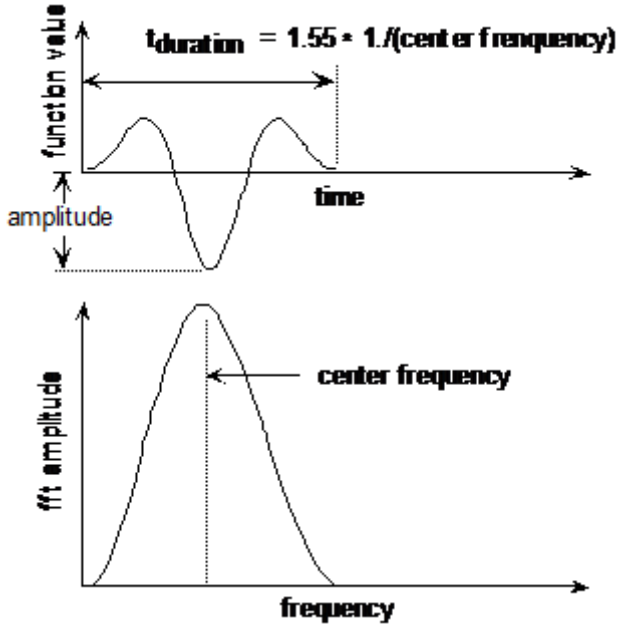
Order Dependence within **FUNC** Group: None.

PCOM	SCOM	input parameters
	BLAK	*centerfrequency amplitude tdelay*

centerfrequency = (F) Center frequency as shown below. No default.

amplitude = (F) Peak amplitude of wavelet. Default = 1.

tdelay = (F) Time delay. Default = 0.0. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.



CHIRP Subcommand
FUNC (function) Input Group **FUNC-CHIRP**

Purpose: To define function parameters for a chirp function.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **FUNC** Group: None.

PCOM	SCOM	input parameters
	CHIRP	*amplitude timeshift frq1 frq2 duration rampup rampdown*

amplitude = (F) Peak-to-peak amplitude of chirp function. No Default.

timeshift = (F) Time shift to beginning time of chirp function. Default = 0.0. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.

frq1, frq2 = (F) Starting and ending frequencies of function. No Default.

duration = (F) Time duration of the chirp function. Must be positive. No Default.

rampup = (F) Time duration to ramp up to full amplitude at start of function. Default= 0.

rampdown = (F) Time duration to ramp down to zero at end of function. Default= 0.

Note: The following equation is used to define the value of the CHIRP function

$f(t) = (ramp * amplitude / 2.) * \sin (2. * \pi * (frq1 + ((chirp / 2.) * time)) * time)$

Where, $chirp = (frq2 - frq1) / duration$

$time = simulation\ time - timeshift$

GAUS (Gaussian) Subcommand
FUNC (function) Input Group **FUNC-GAUS**

Purpose: To define the function to be a two-sided Gaussian time history.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **FUNC** Group: None.

PCOM SCOM input parameters

GAUS *width amplitude tdelay*

width = (F) Time width between the positive and negative peaks of the two-sided Gaussian distribution. Approximately equal to one half the total pulse duration.
Default = 1.0.

amplitude = (F) Amplitude of the Gaussian pulse.
Default = 1.0.

tdelay = (F) Time delay. Default = 0.0. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.

HIST (history) Subcommand
FUNC (function) Input Group **FUNC-HIST**

Purpose: To define the function to be a digitized time history.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **FUNC** Group: None.

PCOM SCOM input parameters

HIST *histname scal adopt tdelay*

histname = (C) Name of the digitized time history previously defined by a **DATA HIST** command.
No default.

scal = (F) Scale factor. Default = 1.0

addopt = (C) If set to ADD, the digitized time history is added to the existing FUNC specification.

tdelay = (F) Time delay. Default = 0.0. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.

SINE (sine wave) Subcommand
FUNC (function) Input Group **FUNC-SINE**

Purpose: To define function parameters for a sine wave.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **FUNC** Group: None.

PCOM SCOM input parameters

SINE *frequency amplitude phaseshift nperiod valuadd rampcycle tdelay*

frequency = (F) Frequency of sine wave in cycles/unit time (in model units). No default.

amplitude = (F) Amplitude of sine wave. No default.

phaseshift = (F) Phase shift of sine wave in degrees. Default = 0.0

nperiod = (F) Length of time (in periods) for sine function to be active. Default or 0.0 produces a continuous sinusoid.

valuadd = (F) Value to add to the sinusoidal value in order to provide a baseline shift to the requested function. Default = 0.

rampcycle= (F) Option to provide a smooth startup for sinusoidal problems. The number of cycles over which the amplitude envelope is smoothly increased from 0.0 to 1.0 is specified. See note 1 below. Default= 0.

tdelay = (F) Time delay. Default = 0, or enter DELAY to inactivate FUNC until it is reset by the TDELAY subcommand.

Note: The following equation is used to define the value of the function for the SIN option over the requested number of periods. Outside this time range, $f(t) = 0.0$.

for $0 \le t \le nperiod*(1/frequenc)$

where, $\omega = 2\pi(frequenc)$ and

for $0 \le t \le rampcycle / frequency$

otherwise $ramp = 1.0$

STEP Subcommand
FUNC (function) Input Group **FUNC-STEP**

Purpose: To define function parameters for a step function.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **FUNC** Group: None.

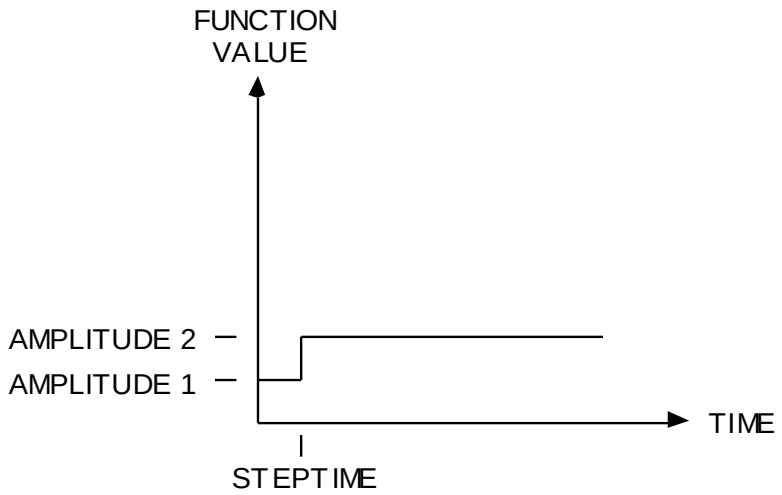
PCOM	SCOM	input parameters
	STEP	*amplitude2 steptime amplitude1 tdelay*

amplitude2 = (F) Amplitude of step function for time > *steptime*.
Default = 0.0

steptime = (F) Time at which step occurs. Default = 0.0

amplitude1 = (F) Amplitude of step function for time ≤ *steptime*.
Default = 0.0

tdelay = (F) Time delay. Default = 0.0. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.



TDELAY (time delay) Subcommand
FUNC (function) Input Group **FUNC-TDELAY**

Purpose: To reset the time delay.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FUNC** Group: None.

PCOM SCOM input parameters

TDELAY *timedelay*

timedelay = (F) Enter new time delay, or RESUME to set *timedelay* to the current time.
Default = 0.

WVLT (wavelet) Subcommand
FUNC (function) Input Group **FUNC-WVLT**

Purpose: To define function parameters for a Rickert wavelet.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **FUNC** Group: None.

PCOM	SCOM	input parameters
	WVLT	*frequency amplitude gamma nu tdelay*

frequency = (F) Frequency of wavelet in cycles/unit time (in model units). No default.

amplitude = (F) Peak amplitude of wavelet. No default.

gamma = (F) Number of subwavelets (typically set to 4.0). Default = 4.0

nu = (F) Phase shift of wavelet (typically set to 0.0). Default = 0.0

tdelay = (F) Time delay of wavelet in periods. Default = *gamma* / 2. Or enter DELAY to inactivate FUNC until it is reset by a TDELAY subcommand.

GAGE Input Command

GAGE

Purpose: To define a set of gage points in continuum elements. Currently only the closest point to each specified location is recorded. Interpolation will be implemented at a future date.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow **GRID** command and precede **PRCS** command.

PCOM	SCOM	input parameters
GAGE		
	LOC	*label array xgage ygage zgage*
	LINE	*label array xbeg xend ybeg yend (zbeg zend) npoints*
	BLOK	*label array axis xbeg xend ybeg yend (zbeg zend) nx ny nz*
	CYLN	*label array axis zbeg zend xcent ycent radbg radend thetabeg thetaend & nr ntheta nz*
	SPHR	*label array xcent ycent zcent radbg radend thetabeg thetaend phibeg & phiend nr ntheta nphi*
	END	

Notes: 1. The GAGE array contains current value at the gage point. Gage labels are in idgg. Gage coordinates are in xgg, ygg, zgg.

2. The POUT HISTNAME command may be used to select a group of gage points for time histories. Either ALL or the *label* may be specified.

LOC Subcommand
GAGE Input Group

GAGE LOC

Purpose: To specify a single gage point

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GAGE** Group: None.

PCOM	SCOM	input parameters
	LOC	*label array xgage ygage zgage*

label = (C) A name associated with this gage point, or set of points. No default.

array = (C) The array to be sampled at the requested location. No default.

xgage, ygage, zgage = (F) Gage point coordinates. Default = (0,0,0)

LINE Subcommand
GAGE Input Group

GAGE LINE

Purpose: To specify a line of gage points

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GAGE** Group: None.

PCOM	SCOM	input parameters
	LINE	*label array xbeg xend ybeg yend (zbeg zend) npoints*

label = (C) A name associated with this gage point, or set of points. No default.

array = (C) The array to be sampled at the requested location. No default.

xbeg…zend = (F) Endpoints of line. Input zbeg zend only for 3D models.

npoints = (I) Number of gage points along line. Default = 2 (the endpoints)

BLOK Subcommand
GAGE Input Group

GAGE BLOK

Purpose: To specify a rectangular region of gage points.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GAGE** Group: None.

PCOM SCOM input parameters

BLOK *label array axis xbeg xend ybeg yend (zbeg zend) nx ny nz*

label = (C) A name associated with this set of gage points. No default.

array = (C) The array to be sampled at the requested location. No default.

axis = (C) Local axis system for BLOK. Default = STND (global). BLOK dimensions are input in local coordinates.

xbeg,...zend = (F) Bounding corners of BLOK. Can be a plane. Only enter *zbeg, zend* for 3D models.

nx,ny,nz = (I) Number of gage points in local x', y', z' directions

CYLN Subcommand
GAGE Input Group

Purpose: To specify a cylindrical region or surface of gage points.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GAGE** Group: None.

PCOM SCOM input parameters

CYLN *label array axis zbeg zend xcent ycent radbg radend thetabeg thetaend & nr ntheta nz*

label = (C) A name associated with this set of gage points. No default.

array = (C) The array to be sampled at the requested location. No default.

axis = (C) Local axis system for BLOK. Default = STND (global). BLOK dimensions are input in local coordinates.

zbeg, zend = (F) Axial extent of cylinder in local axis system. Default = 0.

xcent, ycent = (F) (x', y') coordinates of cylinder axis in local coordinates.

radbg, radend = (F) radial extent of cylindrical region. Can be a single surface.

thetabeg, thetaend = (F) angular extent of cylindrical region. See AXIS primary command for a sketch.

nr, ntheta, nz = (I) Number of gage points in r, theta and z directions

SPHR Subcommand
GAGE Input Group

GAGE SPHR

Purpose: To specify a spherical region or surface of gage points.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GAGE** Group: None.

PCOM SCOM input parameters

SPHR *label array xcent ycent zcent radbg radend thetabeg thetaend phibeg & phiend nr ntheta nphi*

label = (C) A name associated with this set of gage points. No default.

array = (C) The array to be sampled at the requested location. No default.

xcent, ycent,zcent = (F) coordinates of sphere center.

radbg, radend = (F) radial extent of spherical region. Can be a single surface.

thetabeg, thetaend = (F) angular extent (from z) of spherical region. See AXIS primary command for a sketch.

phibeg, phiend = (F) angular extent (in x,y plane) of spherical region. See AXIS primary command for a sketch.

nr, ntheta, nphi = (I) Number of gage points in *r, theta* and *phi* directions

GEOM (geometry) Input Command

Purpose: To define nodal coordinates for the model.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence:Must follow **GRID** and precede **MODL** command.

PCOM	SCOM	input parameters
GEOM		
	XCRD	*xbegin xend ibegin iend ratio*
	YCRD	*ybegin yend jbegin jend ratio*
	ZCRD	*zbegin zend kbegin kend ratio*
	KEYPNT	*nikey njkey nkkey*
	BEND	*axisname neutral_x neutral_y neutral_z ibgin iend jbgin jend kbgin kend*
	FILE	*infilename*
	NCHK	
	SKEW	*(form) ibegin iend jbegin jend kbegin kend*
	GCON	*nodegc x-coord_local y-coord_local z-coord_local (groupnames)*
	PRNT	
	OUT	*form*
	END	

XCRD (X-coordinate) Subcommand
GEOM (geometry) Input Group **GEOM-XCRD**

Purpose: To define grid discretization along the x-axis for standard partition nodes.

Use Is: Required if using a standard partition.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: Typically none.

PCOM	SCOM	input parameters
	XCRD	*xbegin xend ibegin iend ratio*

xbegin, xend = (F) Beginning and ending x-coordinates bounding a section of the grid within which uniform discretization exist.
No defaults.

ibegin, iend = (I) Beginning and ending nodal I-indices of the grid section defined by *xbegin* and *xend*.
Default: *ibegin* = 1, *iend* = maximum I

ratio = (F) Size ratio of succeeding elements in the x-direction for elements of the the grid section bounded by *xbegin* and *xend*.
Default = 1.0

YCRD (Y-coordinate) Subcommand
GEOM (geometry) Input Group **GEOM-YCRD**

Purpose: To define grid discretization along the y-axis for standard partition nodes.

Use Is: Required for 2D and 3D models if using a standard partition.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: Typically none

PCOM SCOM input parameters

YCRD *ybegin yend jbegin jend ratio*

ybegin, yend = (F) Beginning and ending y-coordinates bounding a section of the grid within which uniform discretization exist.
No defaults.

jbegin, jend = (I) Beginning and ending nodal J-indices of the grid section defined by *ybegin* and *yend*.
Default: *jbegin* = 1, *jend* = maximum J

ratio = (F) Size ratio of succeeding elements in the y-direction for elements of the the grid section bounded by *ybegin* and *yend*.
Default = 1.0

ZCRD (Z-coordinate) Subcommand
GEOM (geometry) Input Group **GEOM-ZCRD**

Purpose: To define grid discretization along the z-axis for standard partition nodes.

Use Is: Required for 3D models if using a standard partition.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: Typically none.

PCOM SCOM input parameters

ZCRD *zbegin zend kbegin kend ratio*

zbegin, zend = (F) Beginning and ending z-coordinates bounding a section of the grid within which uniform discretization exist.
No defaults.

kbegin, kend = (I) Beginning and ending nodal K-indices of the grid section defined by *zbegin* and *zend*.
Default: *kbegin* = 1, *kend* = maximum K

ratio = (F) Size ratio of succeeding elements in the z-direction for elements of the the grid section bounded by *zbegin* and *zend*.
Default = 1.0

KEYPNT (Key Point) Subcommand
GEOM (Geometry) Input Group **GEOM-KEYPNT**

Purpose: A shortcut for standard partition models. This option computes the coordinates based on user defined symbols for coordinates and spacings.

Use Is: Optional.

Multiple Use: Not applicable.

Order Dependence
Within **GEOM** Group: None.

PCOM	SCOM	input parameters
	KEYPNT	*nikey njkey nkkey*

nikey = (I) Number of keypoints defined in the I direction. See Note 1 below. No defaults.

njkey = (I) Number of keypoints defined in the J direction. See Note 1 below. No defaults.

nkkey = (I) Number of keypoints defined in the K direction. (3D models only) See Note 1 below. No defaults.

Notes: 1. This option provides a shortcut to multipleXCRD,YCRD, ZCRD commands. The KEYPOINT indices are defined by symbols $I1, $I2, $I3,... $J1, $J2, $J3 etc. The keypoint coordinates are define by symbols $X1, $X2, $X3,... $Y1, $Y2, $Y3, etc.
For example, if *nikey* = 4; it would be equivalent to:

```
XCRD $I1 $I2 $X1 $X2
XCRD $I2 $I3 $X2 $X3
XCRD $I3 $I4 $X3 $X4
```

BEND Subcommand
GEOM (Geometry) Input Group **GEOM-BEND**

Purpose: Curves a region of mesh about an axis of revolution in cylindrical coordinates or about a center-point in spherical coordinates. NOTE: this option must be used in conjunction with a skewed grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GEOM** Group: Must follow the SKEW subcommand.

PCOM SCOM input parameters

BEND *axisname neutral_x neutral_y neutral_z ibgin iend jbgin jend kbgin kend*

axisname = (C) Name of previously defined spherical or cylindrical axis system. No Default.

neutral_x, neutral_y, neutral_z = (F) X,Y, Z coordinates of neutral point, whose coordinates will remain unchanged. No default.

ibgin, iend = (I) The beginning and ending I-indices of nodes which bound the skewed partition.
Default: *ibegin* = 1, *iend* = maximum I

jbgin, jend = (I) The beginning and ending J-indices of nodes which bound the skewed partition.
Default: *jbegin* = 1, *jend* = maximum J

kbgin, kend =(I) The beginning and ending K-indices of nodes which bound the skewed partition.
Default: *kbegin* = 1, *kend* = maximum K

Notes:
1. If a cylindrical coordinate system is used the z' axis defines the axis of revolution.
2. If a spherical coordinate system is used the origin defines the center of curvature.
3. The radius of curvature is defined by the distance from the local coordinate system origin to the neutral point. The neutral point cannot be coincident with the origin.
4. The neutral point is typically within the mesh to define the neutral surface of the curved body, but it can be anywhere in space. Equally spaced points on the neutral surface (perpendicular to radius vector and passing through the neutral point) remain equidistant after bending.
5. Note that using a very small radius of curvature can lead to a highly warped mesh and negative volume elements.

NCHK (No check) Subcommand
GEOM (geometry) Input Group **GEOM-NCHK**

Purpose: For standard partition models, i.e., models with no skewed partition, this option deactivates checking of grid coordinates that ensures the coordinates are monotonically increasing with nodal index.

Use Is: Optional.

Multiple Use: Not applicable.

Order Dependence within **GEOM** Group: None.

PCOM	SCOM	input parameters
	NCHK	

Notes: 1. By default for standard partition models, the grid topology is assumed to be X(i+1)>X(i), Y(j+1)>Y(j), and Z(k+1)>Z(k). The grid geometry is checked to ensure that these constraints are satisfied. If not, PZFlex issues a fatal error and terminates processing.
2. If the **NCHK** subcommand is input, the grid topology is not checked.

FILE Subcommand
GEOM (geometry) Input Group **GEOM-FILE**

Purpose: To define the name of the Grid Input File with input commands.

Use Is: Optional. The name of this file may be default and can also be specified on the program execution command line.

Multiple Use: Not allowed.

Order Dependence within **GEOM** Group: Must precede the **SKEW** subcommand which reads coordinates form the file.

PCOM SCOM input parameters

FILE *infilename*

infilename = (C) Name defining the Grid Input File. May be up to 80 characters long. This file contains coordinates for the skewed partition of the model.

SKEW Subcommand
GEOM (geometry) Input Group **GEOM-SKEW**

Purpose: To define the coordinates of all nodes contained within the skewed partition, including nodes in common with the standard partition. The coordinates are read from the Grid Input File unless the LOCL or STND form option is used.

Use Is: Required for models which have skewed partitions.

Multiple Use: Not allowed.

Order Dependence within **GEOM** Group: None.

PCOM SCOM input parameters

SKEW *(form) ibegin iend jbegin jend kbegin kend*

form = (C) Optional input parameter. The form of the coordinate data.
Any of: BIN= binary data file
FRMT= formatted data file
LOCL= formatted data read from local job input file immediately following the **SKEW** subcommand
STND= create skewed coordinates from standard partition coordinate definition
GCON= skewed node coordinates will be defined for nodes in the ijk-window assigned to the current general connectivity mapping (see the **GCON** command group) by using the "**GEOM GCON**" subcommand. Any following *ibegin, iend*, etc. parameters input are ignored.
Default = BIN.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the skewed partition.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the skewed partition.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the skewed partition.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. The data structure of the Grid Input File is described in the "Files and Formats" section.
2. It is recommended that the binary form for the Grid Input File be used for large files due to the overhead associated with reading formatted text data files with the free format interpreter. Reading a binary file can be an order of magnitude faster.

GCON (general connectivity) Subcommand
GEOM (geometry) Input Group **GEOM-GCON**

Purpose: To define the coordinates of skewed partition nodes using a "general connectivity" form of node identification in contrast to the typical structured grid ijk specification. See Section 2: "Importing General Connectivity Models"

Use Is: Optional. This is a simple way to import a general connectivity model into the computational mesh.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: None.

PCOM SCOM input parameters

GCON *nodegc x-coord_local y-coord_local z-coord_local (groupnames)*

nodegc = (I) Local general connectivity node number. The node number is remapped to the current grid based on the current general connectivity mapping specified by the last **GCON SET** command.
No default.

x-coord_local, y-coord_local, z-coord_local = (F) x-, y-, and z-coordinates of the node specified in the local system of the current general connectivity definition. The coordinate is scaled and transformed according to the transformes defined in the **GCON** command group.
Default: 0.0

groupnames =(C) A set of group names can be appended to the end of this subcommand. Information about nodes of a particular group can be accessed using the **SYMB** #GET GCON option.

PRNT (print) Subcommand
GEOM (geometry) Input Group **GEOM-PRNT**

Purpose: Print grid discretization information.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: None.

PCOM	SCOM	input parameters
	PRNT	-none-

OUT Subcommand
GEOM (geometry) Input Group **GEOM-OUT**

Purpose: To write the grid coordinates to the Grid Output File.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **GEOM** Group: Should follow all coordinate definition subcommands.

PCOM SCOM input parameters

OUT *form*

form = (C) Form of the Grid Output File.
Any of: FRMT = formatted text file
BIN = binary file

Default = BIN

GLUE Input Command

Purpose: To define surfaces of the grid that are glued together. Although the surfaces may have different discretizations, accuracy is better if the discretization of the more finely discretized surface is an integer multiple of the discretization of the coarser surface (1 to 1 is best). Nodes on the coarse surface are master nodes, nodes on the fine surface are slave nodes.
GLUE is a more generalized version of BOND.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow **GRID** command and precede **PRCS** command.

PCOM SCOM input parameters

GLUE

DEFN	*gluename keyword option*
MASTER	*(option) ibegin iend jbegin jend kbegin kend*
MASTER	*gcoption n1 n2 (n3 n4 n5 n6 n7 n8)*
SLAVE	*(option) ibegin iend jbegin jend kbegin kend*
SLAVE	*gcoption n1 n2 (n3 n4 n5 n6 n7 n8)*
END	

Note: If kinematic boundary conditions are applied to glued interface nodes, they must be applied to the master nodes.

DEFN (definition) Subcommand
GLUE Input Group **GLUE-DEFN**

Purpose: To begin the definition of a new glued interface.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
within **GLUE** Group: Must precede the **MASTER** and **SLAVE** subcommands that specify which surfaces are to be glued together.

PCOM	SCOM	input parameters
	DEFN	*gluename keyword option*

gluename = (C) Unique name associated with the glued interface being defined.

keyword = (C) If set to NCHK, option may be set to X, Y, or Z to glue surfaces that are physically separated in a Cartesian direction. If set to FUZZ, the relative fraction of master element size allowed as an offset for glued surfaces (default = 0). If the FUZZ option is not specified, the tolerance is 0.01. Input a negative value of FUZZ for an absolute tolerance.
Default: only contacting surfaces are glued.

Notes:
1. When master segments define three-dimensional volumes, slave nodes must fall within the volume (or on the boundary to be glued). If slave nodes are glued to a surface the FUZZ option can be used to allow an offset.
2. When using the NCHK option, care should be taken not to specify a Cartesian direction parallel to the master surface. Doing so can result in slave nodes being glued to non-adjacent segments on the master surface.

MASTER (master surface) Subcommand
GLUE Input Group **GLUE-MASTER**

Purpose: To define the more coarsely discretized surface being glued.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
within **GLUE** Group: Must occur after the appropriate **DEFN** subcommand.

PCOM SCOM input parameters

MASTER *(option) ibegin iend jbegin jend kbegin kend*
or
MASTER *gcoption n1 n2 (n3 n4 n5 n6 n7 n8) (gcname)*

option = (C) ijk option
IJK: linear, planar or volume ijk-defined Segment. If omitted or defaulted *option* = IJK

gcoption = (C) GCON option. Enter one of the following to indicate the segment type:
LINE: 2-node linear segment
SURF: 4-node planar surface (*n4=n3* for triangles)
TET: 4-node tetrahedral volume
HEX: 8-node hexahedral volume

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the master region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the master region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the master region.
Default: *kbegin* = 1, *kend* = maximum K

n1,n2,...,n8 = (I) The general connectivity node numbers defining the segment type indicated in *option.* Enter two nodes for LINE, 4 nodes for SURF and TET, 8 nodes for HEX. Nodes must be entered in counter-clockwise fashion to properly define the segment.

gcname = (C) optional gcon definition. Default as specified on last USE subcommand.

Notes: 1. As the glued surface is assumed to be a plane in IJK space, *ibegin = iend*, *jbegin = jend*, or *kbegin = kend*.
2. Glue master surfaces can be visualized by plotting the GLUE_MASTER array.

SLAVE (slave surface) Subcommand
GLUE Input Group **GLUE-SLAVE**

Purpose: To define the more finely discretized surface being glued.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
within **GLUE** Group: Must occur after the appropriate **DEFN** subcommand.

PCOM SCOM input parameters

SLAVE *(option) ibegin iend jbegin jend kbegin kend*
or
SLAVE *gcoption n1 n2 (n3 n4 n5 n6 n7 n8) (gcname)*

option = (C) ijk option.
IJK: linear, planar or volume ijk-defined segment
If omitted or defaulted *option* = IJK.

gcoption = (C) Optional input. Enter one of the following to indicate the segment type:
NODE: single node
LINE: 2-node linear segment
SURF: 4-node planar surface (*n4=n3* for triangles)
TET: 4-node tetrahedral volume
HEX: 8-node hexahedral volume

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the slave region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the slave region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the slave region.
Default: *kbegin* = 1, *kend* = maximum K

n1,n2,...,n8 = (I) The general connectivity node numbers defining the segment type indicated in *option.* Enter two nodes for LINE, 4 nodes for SURF and TET, 8 nodes for HEX.

gcname = (C) optional gcon definition. Default as specified on last USE subcommand.

Notes: 1. As the glued surface is assumed to be a plane in IJK space, *ibegin = iend, jbegin = jend,* or *kbegin = kend.*
2. Glued slave nodes can be visualized by plotting the GLUE_SLAVE array.
3. If slave nodes that do not fall on a master surface within the same glue definition are assigned, the nodes are not glued. These nodes can be visualized by plotting the UNGLUED array.

GRID Input Command

GRID

Purpose: To define the number of nodes in the computational grid, the number of dimensions for the model, and the appropriate constraint relations.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence:Must precede **GEOM** command.

PCOM input parameters

GRID *igrid (jgrid) (kgrid) constraint*

igrid = (I) Range of I-indices for the grid. No default.

jgrid = (I) Optional input. Range of J-indices for the grid. Input only for 2D and 3D models. No default.

kgrid = (I) Optional input. Range of K-indices for the grid. Input only for 3D models. No default.

constraint = (C) Constraint relations for model.

For 1D models:
Any of: AXIL = axial wave propagation
SH = shear wave propagation (elastic only)
Default = AXIL

For 2D models:
Any of: PSTN = plane strain
AXIX = axisymmetric about x-axis
AXIY = axisymmetric about y-axis
SH = sh-wave model (elastic only)
Default = PSTN

For 3D models: Leave blank.

Note: 2D PSTN and SH models assume a model that is a unit thickness in and out of plane. AXIX and AXIY models assume a one-radian arc model.

Examples: The following examples illustrate the use of the GRID command.

GRID 50 (1D model, axial wave propagation with 50 nodes)
GRID 100 40 AXIX (2D axisymmetric model, radius is y-axis)
GRID 30 60 (2D plane strain model, 30 x 60 grid nodes)
GRID 400 200 100 (3D model, 400 x 200 x 100 = 8,000,000 nodes)

GRPH (graphics) Input Command

GRPH

Purpose:To provide interactive graphics display of the computational model and results of an analysis. Also provides for interactive or batch mode generation of PostScript image files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the **GEOM** and **SITE** commands to plot continuum elements. If plotting shell elements, it must follow the **SHEL** command group. If plotting **BM3D** elements, it must follow the **PRCS** command group.

PCOM	SCOM	input parameters
GRPH		
	ACTV	*type option*
	ARROW	*option scale nspace (options) ibegin iend jbegin jend kbegin kend*
	BBOX	*type (xmin xmax ymin ymax zmin zmax)*
	BLOK	*ibegin iend jbegin jend kbegin kend*
	CLER	*type*
	CLOS	
	COLR	*(option) ival tred tgreen tblue hred hgreen hblue*
	COMB	*newname oldname1 oldname2 ... oldnamen*
	DISP	*xscale yscale zscale*
	DRAW	NODE *ibegin iend jbegin jend kbegin kend*
	DRAW	CORD *xbegin ybegin zbegin xend yend zend*
	DUPL	*xshift yshift zshift*
	EDGE	
	EYE	*xvalue yvalue zvalue*
	FRAME	*x-space y-space z-space option*
	IMAG	
	INVR	
	LINE	*option*
	MAP	*materialname colorindex*
	MIRR	*axis option dataoption*
	NODE	*option*

PCOM SCOM input parameters

NVEW	*nview*
PAN	*pan_option zoom_value (all_option) (save_option)*
PLOT	(*ijk_option*) MATR (BLOK *blockname*)
PLOT	[*datan*] (BLOK *blockname*) (RANG *minvalue maxvalue*)
PLOT	SHAP *numshape* (*shapdata*) *phaseshift*
PLOT	FLXHST *irecordn*
PS	*option*
REVR	*iaxis option*
SET	*option parameters*
SIZE	*xsize ysize xbegin xend ybegin yend nxdiv nydiv*
SWAP	*option*
TIME	*itimestep time*
TTL	*ititle*

<----------200-character title information for line = *ititle*---------->

VERT	*xvalue yvalue zvalue*
VIEW	*iview*
VPNT	*xvalue yvalue zvalue*
WNDO	(option1) (option2) *xbegin xend ybegin yend zbegin zend*
WRIT	(*options*) *filename*
WSIZ	*width height*
ZOOM	*zoom_value (all_option) (save_option)*

ACTV (active) Subcommand
GRPH (graphics) Input Group

GRPH-ACTV
(3D models only)

Purpose: To activate or deactivate the continuum or shell portions of the model for plotting.

Use Is: Optional. Default is that all shell and continuum elements are active.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	ACTV	*type option*

type = (C) Element type being activated or deactivated.
Any of: CONT - continuum element
SHEL - shell elements
BEAM - beam elements (BM3D elements for material plotting only)
No default.

option = (C) Control option.
Any of: YES - element type becomes active
NO - element type becomes inactive
Default: YES

Notes: 1. If continuum elements are inactive, they are not plotted.
2. If shell elements or beam elements are inactive, they are not plotted.

ARROW Subcommand
GRPH (graphics) Input Group **GRPH-ARROW**

Purpose: To plot arrows representing various vector data.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **GRPH** Group: Applies to all following **PLOT** subcommands.

PCOM SCOM input parameters

ARROW *option scale nspace ibegin iend jbegin jend kbegin kend*
or
ARROW CSTM *scale nspace arrwx arrwy (arrwz) iend jbegin jend kbegin kend*

option = (C) Set to POLE, CSTM, or OFF.
POLE: plots material poling direction (dielectric orientation) for piezoelectric materials.
SHEL: plots shell element directions
CSTM: plots custom vector data given the x, y and z vector component arrays
MATR: plots continuum material directions
OFF: clears all arrows for subsequent PLOT commands
No default.

scale = (F) Scale factor for arrow size.
Default = 10 * element size.

nspace = (I) Element spacing between arrows.
Default = 10

options = (X) For *type* = POLE:
No options.

For *type* = SHEL:
option1 = *shelval*
option2 = *arrayname*

shelval = (C) Shell value to be plotted. Enter D1 to plot the shell local x (1) direction. Enter D2 to plot the shell local y (2) direction. Enter D3 to plot the shell local z (3) direction (ie. element normal). Enter PTEN to plot principle stress or strain directions.

arrayname =(C)Required only if *option1* = PTEN. Name of data array containing principle values from CALC PTEN command.

ARROW Subcommand
GRPH (Graphics) Input Group

GRPH-ARROW
(continued)

PCOM SCOM input parameters

For *type* = CSTM:
option1 = *arrwx*
option2 = *arrwy*
(option3 = *arrwz)*

arrwx, arrwy, arrwz = (C) Names of x, y and z vector component arrays.

For *type* = MATR:
option1 = *matval*

matval = (C) Material direction to be plotted. Enter D1 to plot the material local x (1) direction. Enter D2 to plot the material local y (2) direction. Enter D3 to plot the material local z (3) direction.

ibegin, …, kend = (I) Nodal indices bounding the region on which to plot arrows. Defaults are model boundaries.

matname = (C) Optional entry to plot arrows only for a given material name (continuums) or layer definition name (shells). No default.

Notes:
1. Poling arrows are only plotted on materials with *mattyp* = LEAN.
2. Only one set of arrows can be plotted at a time. Subsequent GRPH ARROW commands overwrite the previous arrows.
3. Examples:

ARROW POLE 10 5
Generates arrows indicating piezo-electric poling directions on all elements of mattyp = LEAN in the model. The arrows are ten times the element size long and spaced every fifth element.

ARROW SHEL 0.5 2 D1 1 10 1 10 1 10
Generates arrows indicating shell element local x (1) directions on all shell elements in the range. Arrows are half an element size long and spaced every second element.

ARROW SHEL 0.5 2 PTEN ptstrs
Generates arrows indicating the principle tension directions from the array 'ptstrs' that was created using the CALC PTEN command. Note that these directions are a snapshot at the time the CALC PTEN command was issued. The CALC PTEN command and the GRPH ARROW command need to be reissued to update the arrows at subsequent time steps. Principle stress and strain directions are plotted without arrowheads.

ARROW CSTM 0.5 2 xvel yvel zvel 1 10 1 10 1 10
Generates velocity vectors on all continuum elements in the indicated range.

ARROW MATR 2. 2 D1 ** ** ** mat1
Generates arrows indicating the material 1-direction for all continuum elements in the model with material definition named mat1.

BBOX (bounding box) Subcommand
GRPH (graphics) Input Group **GRPH-BBOX**

Purpose: Option to control the plotting of a bounding box when plotting model or snapshot data.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Applies to all following **PLOT** subcommands.

PCOM SCOM input parameters

BBOX *type (xmin xmax ymin ymax zmin zmax)*

type = (C) Bounding box type. Any of:
USER = user-defined bounding box
PNBL = pinball bins bounding box
OFF = turn off bounding box plot
No Default.

xmin, xmax, ymin, etc. = (F) For type = USER. Minimum and maximum bounds of user-defined bounding box.
No Default.

BLOK (block) Subcommand
GRPH (graphics) Input Group **GRPH-BLOK**

Purpose: To define a blocking construct that defines a subset of the grid for plotting

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

(general form)

BLOK *(blockname) (option) [parameters]*

blockname = (C) Optional input. Name of the blocking definition. If not input, *blockname* = B1 is assumed.

option = (C) Optional input. Blocking option.
Any of: INDX, RANG, CLER
INDX - a set of nodal indices defining the block is provided. Multiple INDX option **BLOK** subcommands for the same *blockname* are accumulative.
RANG -a set of data value ranges is used to block the data for plotting. Multiple RANG option **BLOK** subcommands for the same *blockname* replace the previous definition.
CLER - all or part of the blocking specification is to be cleared
Default = INDX

[parameters] = (X) A set of input parameters whose type and function are dependent on *option* as described below.

(specific forms)

BLOK (*blockname*) INDX *ibegin iend jbegin jend kbegin kend*

BLOK (*blockname*) RANG *vbegin1 vend1 vbegin2 vend2 ... vbeginn vendn*

BLOK (*blockname*) CLER *clearoption*

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the region to plot.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the region to plot.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the region to plot.
Default: *kbegin* = 1, *kend* = maximum K

vbeginn, vendn = (F) Pairs of values that define data ranges for elements to be plotted. If an element's data value falls between vbeginn and vendn, the element is plotted, otherwise it is not. No defaults.

clearoption = (C) Clear option for blocks. Any of: INDX, RANG, ALL
INDX - clears all indices blocks for *blockname*
RANG - clears all data range blocks for *blockname*
ALL - clears both indices and range blocks for *blockname*
Default=ALL

Note: The following commands set up a blocking construct named B2, plot a data field DAT1 with the blocking applied, and then clear the B2 blocking definition.

```
BLOK   B2  1 20 10 50 1 4
BLOK   B2 RANG -100. -50. 50. 100.
PLOT   DAT1 BLOK B2
BLOK   B2 CLER
```

CLER (clear) Subcommand
GRPH (graphics) Input Group **GRPH-CLER**

Purpose: To clear the current group of continuum plotting blocks (defined with the **BLOK** subcommand) and reset plotting for the entire grid. Alternatively, can be used to clear previously defined shell blocks or draw definitions.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

CLER *type*

type = (C) Type of elements for which previously defined blocks will be cleared.
Any of: CONT - clears continuum blocks
SHEL - clears shell blocks
DRAW - clears draw definitions

Default: Clears blocks for all active continuum and shell blocks. Does not affect draw definitions.

CLOS (close) Subcommand
GRPH (graphics) Input Group **GRPH-CLOS**

Purpose: To close the graphics window.

Use Is: Optional. If not input, the graphics window is closed at the end of the job.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	CLOS	-none-

COLR (color) Subcommand
GRPH (graphics) Input Group **GRPH-COLR**

Purpose: To specify the current color table or the red-green-blue contributions for any of the individual colors of the current table. The modified colors are used in all following plots. Also used to list the rgb values of the current color table.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

COLR TABL *option itable* /* selects a color table for plotting
or
COLR LIST *option* /* lists the rgb values of a color table
or
COLR FORE *red green blue* /* sets the foreground color
or
COLR BACK *red green blue* /* sets the background color
or
COLR USER SIZE *ncolors* /* sets number of colors for user table
or
COLR USER *icolor red green blue* /* changes a color in the user table

option = (C) Set to MATR to choose material color table or to DATA to choose data color table. Default = DATA.

itable = (I) Color table number to use for plotting *option* type information. From 1 to 8. No default.

ncolors = (I) Number of the colors in table 7 (the user-definable color table). From 2 to 39.
No default.

icolor = (I) Color number in table 7 (the user-definable color table) to set the color for. From 1 to *ncolors*.
No default.

red, green, blue = (F) Red-green-blue values that define a color

Note: There are seven predefined color tables, as shown in Section 2 (Tables 1 to 6 and Table 8). All have 13 colors except for Table 5, which has 39, and Table 6, which has 120. Table 7 is user-definable. The number of colors and individual color definitions can be set.

Example:
```
COLR TABL MATR 5        */ sets color Table 5 for succeeding material plots
COLR TABL DATA 6        /* set color Table 6 for succeeding data plots
```

COMB (combine) Subcommand
GRPH (graphics) Input Group **GRPH-COMB**

Purpose: Primarily used to combine elements with different material properties into a single type for plotting with the **PLOT** MATR subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

COMB *(option) datatype newname oldname1 oldname2 ... oldnamen*

option = (C) Any of: EXCEPT, ROOT
EXCEPT — All names will be changed to newname except for those *oldnames* listed.
ROOT — Every name that begins with *newname* will be combined to *newname*
Default = no option

datatype = (C) Identifies the type of data which are being combined for plotting. For material properties, set to MATR.
No default.

newname = (C) New name that applies to all elements having the old names listed. *newname* may be any name that is already used or a name that is not an actual material name.
If *newname* = OFF, then all previous COMB subcommands for this data type are deactivated.
No default.

oldname1,..., oldnamen = (C) List of old names that will be combined to form the new name.
No default.

Notes: 1. If material names are combined into the VOID material name, those portions of the model become void for plotting also.
2. This option is useful to reduce the number of materials down to a reasonable number to display.
3. The **COMB** MATR OFF option allows the user to restore the original material properties to the model.

DISP (displacement) Subcommand
GRPH (graphics) Input Group **GRPH-DISP**

Purpose: To add the current nodal displacements (multiplied by a scale factor) to the grid coordinates for plotting the deformed shape of the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	DISP	*xscale yscale zscale*

xscale = (F) Scale factor for the x-displacements.
Default = 0.0

yscale = (F) Scale factor for the y-displacements.
Default = *xscale*

zscale = (F) Scale factor for the z-displacements (3D models only).
Default = *xscale*

Note: The displacement scale factors input on the **DISP** subcommand are consistent for both small and large deformation simulations. The difference between small and large deformation calculations is that for large deformation simulations, the geometry of the model is constantly updated throughout a simulation, whereas for small deformation simulations it is not. Therefore, if the model is plotted at some advanced time during a simulation without a **DISP** subcommand having been input, a small deformation model plots the undeformed geometry, while a large deformation simulation plots the deformed geometry with a displacement scale factor of 1. This is indicated on the plot. To plot an undeformed shape for a large deformation run, a user can input the **DISP** subcommand with a displacement scale factor of 0.0. This subcommand removes the current displacements from the current coordinates to produce an undeformed model plot.

DRAW Subcommand
GRPH (graphics) Input Group **GRPH-DRAW**

Purpose: Specify lines to draw on the model when plotted. Lines specified are drawn on the graphic generated by any succeeding **PLOT** subcommands. This option is available only for 2D models.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters		
		(NODE type **DRAW** subcommand)		
DRAW		NODE *ibegin iend jbegin jend kbegin kend*		
	ibegin, jbegin, kbegin =	(I)	For NODE type command, the ijk indices of the node at the beginning of the line to be drawn.	
	iend, jend, kend =	(I)	For NODE type command, the ijk indices of the node at the end of the line to be drawn.	
		(CORD type **DRAW** subcommand)		
DRAW		CORD *xbegin ybegin zbegin xend yend zend*		
	xbegin, ybegin, zbegin =	(F)	For CORD type command, the x-, y-, and z-coordinates of the beginning of the line to be drawn.	
	xend, yend, zend =	(F)	For CORD type command, the x-, y-, and z-coordinates of the end of the line to be drawn.	
DRAW		CLER	Clears all previous DRAW specifications.	
DRAW		LINE	Displays all line elements as lines.	

Notes: 1. Any number of **DRAW** subcommands may be specified. They can be any mix of the NODE, CORD, and LINE types.
2. Lines specified by the **DRAW** subcommand are stored in memory and drawn on all plots generated after the **DRAW** subcommands have been input.

DUPL (duplicate) Subcommand
GRPH (graphics) Input Group **GRPH-DUPL**

Purpose: To request duplication of the model plot at a position relative to the actual model's location. The next **PLOT** subcommand displays the original and the duplicated geometry.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

<to duplicate the model geometry>

DUPL *xshift yshift zshift*

xshift,yshift, zshift = (F) Duplicated model is shifted *xshift* distance along the x-axis, *yshift* distance along the y-axis, and *zshift* distance along the z-axis. Default = 0.0

<to clear all previous **DUPL** subcommands>

DUPL CLER

Notes:
1. Multiple **DUPL** subcommands are accumulative.
2. The amount of memory storage required to display large models can be significant. Using one **DUPL** command doubles the memory requirements, using two triples the requirements, etc.
3. The duplicated geometry includes all mirror options currently active.

EDGE Subcommand
GRPH (Graphics) Input Group **GRPH-EDGE**

Purpose: To refresh the material interface and edge lines before plotting with element lines turned off. May be necessary if elements have eroded since previous plot.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GRPH** Group: Must precede PLOT subcommand.

PCOM	SCOM	input parameters
	EDGE	

Notes: 1. Edge recalculation is not done automatically since it may take time for large models.

EYE (eye point) Subcommand
GRPH (graphics) Input Group

GRPH-EYE
(3D models only)

Purpose: To define the eye point used to define the view orientation of the following plots of 3D models. The other components of the viewing orientation are defined by the **VPNT** and **VERT** subcommands.

Use Is: Optional. Default is x=0.0, y=0.0 and z=-1.0.

Multiple Use: Allowed.

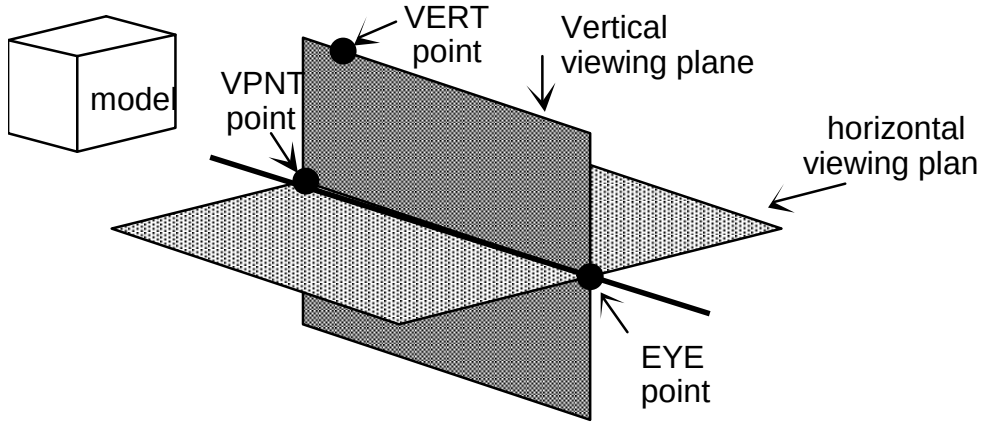
Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	EYE	*xvalue yvalue zvalue*

xvalue = (F) x-coordinate of the eye point.
Default = 0.0.

yvalue = (F) y-coordinate of the eye point.
Default = 0.0.

zvalue = (F) z-coordinate of the eye point.
Default = 0.0.



Definition of viewing orientation for 3D models

FRAME Subcommand
GRPH (Graphics) Input Group **GRPH-FRAME**

Purpose: To plot a background frame to indicate the scale of the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GRPH** Group: Applies to all following **PLOT** subcommands.

PCOM SCOM input parameters

FRAME *x-space y-space z-space option*

x-space, y-space, z-space = (F) The x,y and z spacing of grid lines.
Default is the model dimension divided by 8 in each direction.

option = (F) The extent of the background frame Set to either:
FULL — background frame covers full size of model
LOCAL — background frame only extends to visible parts of plotted model
Default is no frame plotted.

Notes: 1. The command GRPH FRAME OFF will turn off the background frame.

IMAG (image) Subcommand
GRPH (graphics) Input Group **GRPH-IMAG**

Purpose: To save the current graphics image to an image output file. This option can be used to save individual images in the TIFF format or to accumulate individual frames of a movie animation in one of several forms. The **GRPH PSET** IMAG option controls what type of image data are written when the IMAG subcommand is invoked. The default image option produces an animation in Thornton Tomasetti proprietary format that can be played back with the **FILM** command in **Review**.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Must follow the commands that generate the image.

PCOM SCOM input parameters

IMAG *(tiff_filename)*

tiff_filename = (I) Optional filename override if the image type has been set to produce TIFF images. If input, the TIFF image is written to this file name. If not input, the tiff image is written to a sequentially numbered file using PZFlex's standard file-naming convention.

INVR (invert) Subcommand
GRPH (graphics) Input Group **GRPH-INVR**

Purpose: A toggle command to invert the order of the colors in the current color table as they are plotted from bottom to top for the next **PLOT** command. This reverses which ends of the the color range are associated with the minimum and maximum values of the data array being displayed.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	INVR	-none-

LINE Subcommand
GRPH (graphics) Input Group **GRPH-LINE**

Purpose: Option to control the plotting of element lines when plotting the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **GRPH** Group: Applies to all following **PLOT** subcommands.

PCOM	SCOM	input parameters
	LINE	*option (nsub innerfl)*

option = (C) Line plotting option. Any of: YES, NO, NONE or SUB.
option = YES plots all mesh lines
option = NO turns off mesh plotting, leaving material interface lines.
option = NONE turns off all mesh plotting and material interface lines.
option = SUB plots a sparse mesh
Default = YES

nsub = (I) Sub-sampling factor for option = SUB. Default is previous setting. See Note 1.

innerfl = (C) Flag to plot interior lines for 3D models when option = SUB. *innerfl* = "ON" plots interior lines. Default = "OFF." See Note 2.

Notes:
1. The SUB option allows a mesh to plotted sparsely in cases where elements are very small but a grid is desired to show deformation. For example, if *nsub* = 2 gridlines are placed every two elements. If *nsub* is input, the code determines which gridlines to plot and stores the information in a data manager array for subsequent GRPH-PLOT commands. If *nsub* is left blank or defaulted, the code uses the previous value of *nsub* and does not need to recalculate the gridlines.
2. For *innerfl* = ON, interior grid lines are determined and stored, which can lead to long processing times during each GRPH-PLOT command. If *innerfl* = OFF, only exterior grid lines are determined and stored, allowing much faster processing during each GRPH-PLOT command. Default is *innerfl* = OFF. Set to ON if interior sections are desired.

MAP Subcommand
GRPH (graphics) Input Group **GRPH-MAP**

Purpose: To map a material name to a specific color in the color table.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

MAP *materialname colorindex*

materialname = (C) Name of material the user wishes to assign to a specific color. No default.

colorindex = (I) Index defining which color of the color table to assign to *materialname*. No default.

Note: Mapping a material to color index 0 (typically the VOID color index) is not allowed unless the VOID material type has been mapped to a different color index.

MIRR (mirror) Subcommand
GRPH (graphics) Input Group **GRPH-MIRR**

Purpose: To mirror the model image about the x-y, y-z, and/or x-z axes planes. This option allows the user to produce graphics images showing a complete model even though the simulation model made use of symmetry or antisymmetry to reduce the model size.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **GRPH** Group: None.

PCOM SCOM input parameters
<to activate/deactivate a mirror plane for an axis>

MIRR *axis option dataoption*

axis = (C) Name of the axis normal to the mirror plane
May be any of: X, Y, Z, or OFF.
If set to OFF, all previous **MIRR** subcommands are deactivated.
No default.

option = (C) Mirror option to use for *axis*.
May be any of: ON or

SYMM = activate symmetric mirroring of geometry
ASYM = activate antisymmetric mirroring of geometry
OFF = deactivate mirroring

dataoption = (C) Mirror option for the data being displayed.
May be any of: SAME or

SYMM = display the mirrored data values with the same sign as the original (unmirrored) values
CHNG or
ASYM = display the mirrored data values with the opposite sign as the original (unmirrored) values

<to define the local origin (position) of the mirror plane on an axis>

MIRR ORIG *axis origincoord*

axis = (C) Name of axis for which the user is setting the coordinate of the mirror plane
May be any of: X, Y, Z. No default.

origincoord = (F) Coordinate of mirror plane
Default = 0.0

<to allow the user to omit certain quadrants of the mirrored geometry>

MIRR OMIT *omitoption*

omitoption = (C) To define which portions of the geometry to omit. (See Note 4 below for sign terminology.)
May be any of: NONE or
OFF = all mirrored quadrants are displayed
any combination of X- or X+, Y-, or Y+, Z- or Z+.
For example: *omitoption* = X-Y-Z+ , when mirrors are active for the x, y, and z axes, produces a plot of a model with one quadrant out of 8 omitted.

Notes:
1. The mirror plane is always assumed to be at an *axis* coordinate of 0.0 unless the user has used the **MIRR** ORIG subcommand to reset the location of the mirror.
2. If the subcommand **MIRR** Y is input and the location of the Y axis mirror is at coordinate Y_0, all graphics images drawn with Y_1 coordinates are also automatically displayed with coordinate = Y_0 -(Y_1 - Y_0) unless the **MIRR** OMIT option has been used to suppress the Y- quadrant mirror display.
3. When mirroring geometry and field variable data, the user is responsible for defining the type of mirror behavior (symmetric or antisymmetric) for each of these quantities.
4. When the **MIRR** OMIT option is used, by definition, the actual geometry defined in the model is considered to be in the + quadrant no matter the sign of the model coordinates, and the mirrored geometry is considered to be in the negative quadrant.

NODE Subcommand
GRPH (graphics) Input Group **GRPH-NODE**

Purpose: Option to control the plotting of nodal data.

Use Is: Optional. If not input, average values of nodal data for each element are displayed.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Applies to all following **PLOT** subcommands.

PCOM SCOM input parameters

NODE *option*

option = (C) Nodal data plotting option. Any of: AVG, MIN or MAX

AVG = display each element with a color corresponding to the average value of nodal data for all of the element's nodes.

MIN = display each element with a color corresponding to the minimum value of nodal data for all of the element's nodes

MAX = display each element with a color corresponding to the maximum value of nodal data for allof the element's nodes

Default = AVG

NVEW (number of views) Subcommand
GRPH (graphics) Input Group **GRPH-NVEW**

Purpose: Set the number of viewing windows present on the screen at one time.

Use Is: Optional.

Multiple Use: Allowed.

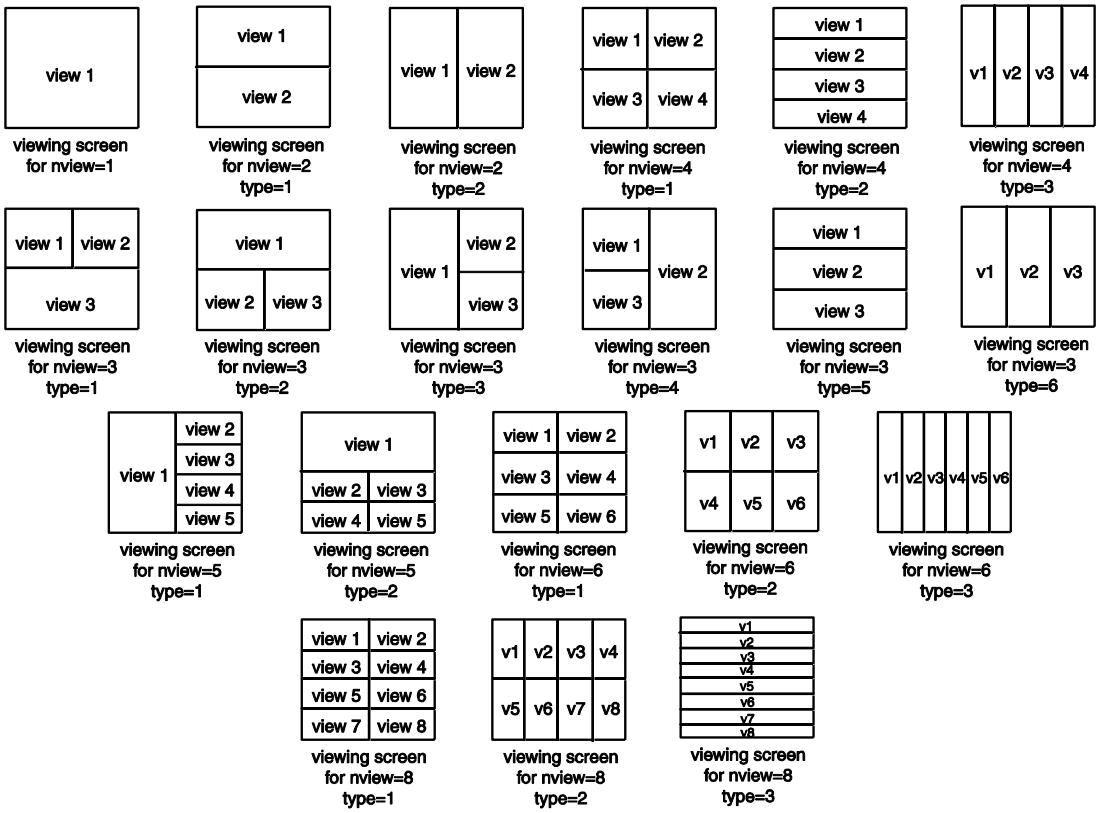
Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

NVEW *nview type*

nview = (I) The number of viewing windows.
Any of: 1, 2, 3, 4, 5, 6, or 8
Default = 1

type = (I) For *nview*=2, 3, 5, 6, or 8, *type* specifies which type of multiview arrangement to use, as shown below.
Default = 1



PAN (image pan) Subcommand
GRPH (graphics) Input Group **GRPH-PAN**

Purpose: Pan the "virtual camera" displaying the model left, right, up, or down. This option in combination with the **ZOOM** subcommand allows for closer inspection of regions of the model currently displayed. The option applies only to images of the model or snapshots of field variables

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

PAN *pan_option zoom_value (all_option) (save_option)*

pan_option = (C) Direction in which to "pan" the "virtual camera" being used to display the model. May be:
L= pan left
R= pan right
U= pan up
D= pan down
C= center camera on the image

No default.

pan_value = (F) Pan distance as a fraction of the overall image. For example, a value of .1 shifts the image about 10% of the initial image size in the pan direction. This parameter is ignored if *pan_option*=C.
Default = 0.0.

all_option = (C) If set to ALL, all current views that contain field plots are redrawn with the panned image. If omitted, this **PAN** subcommand applies only to the current view.
No default.

save_option = (C) If set to SAVE, then the pan setting will be retained for the current view. Can be combined with the ALL option. Default OFF — settings are cleared.

Note: When an image is displayed for the first time, a clipping border is established that bounds the image on the screen. The **PAN** subcommand does not alter this clipping border.

PLOT Subcommand
GRPH (graphics) Input Group **GRPH-PLOT**

Purpose: To graphically display information about the model and its computed results during an analysis. Primary uses include plotting the materials assigned to the mesh, plotting the spatial variation of field variables, plotting the results of extracted shape data at the end of an analysis, and displaying time history data during the analysis.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

<< plot materials assigned to the mesh >>

PLOT (*ijk_option*) MATR (BLOK *blockname*) (BARLABEL *label*)

<< plot spatial variation of field variables in the mesh >>

PLOT [*datan*] (BLOK *blokname*) (RANG *minvalue maxvalue*) (BARLABEL *label*)

<< plot extracted shape results >>

PLOT SHAP *numshape* (*shapdata*) *phaseshift*

<< plot time history records (**POUT** command) up to the current simulation time >>

PLOT FLXHST *irecordn*

ijk_option = (C) Optional input to plot elemental or nodal grid quantities, in ijk- instead of xyz-space. Set to IJK if plotting in ijk-space is preferred. Skip if xyz-space is preferred.

[*datan*] = (X) Defines which field variable to plot. There may be up to 6 data objects on a single **PLOT** subcommand. The construct begins with a data array name (e.g., PRES, EX, etc.) followed by a set of optional plotting parameters. The optional parameters are I, J, K, and VS.. If no optional parameters are input, the entire data array is plotted. The I, J, K parameters allow the user to plot a subset of the requested data. If the data being plotted form an 1D set, the VS. option allows the data to be plotted vs.. the x-, y-, or z-coordinate of the data values if applicable.
See examples below for typical usage.

<<BLOK - optional block definition to apply to the data being displayed >>
If not input, the entire data set is displayed

blokname = (C) Name of the block definition to use. Must have been previously defined by the **BLOK** subcommand
Default = no default

<<BARLABEL - optional color bar label >>
If not input, the default label is displayed

label = (C) label to display

<<RANG - optional input to control the scale range of the color bar>>

If not input, the color bar is auto-scaled to the appropriate range of the requested data. The user can control the color bar scale by using the RANG set of parameters.

minvalue = (C) Minimum data value to use to set the color scale.
Default = minimum data value contained in all *datan.*

maxvalue = (C) Maximum data value to use to set the color scale.
Default = maximum data value contained in all *datan.*

<<SHAP - option to allow plotting extracted steady-state resonance shape data>>

This option is used at the end of a simulation to display information requested by the **SHAP** command group. Note: shape magnification is controlled by the **DISP** subcommand.

numshape= (I) Transform corresponding to frequency *numshape* as declared in the **SHAP** input group.

shapdata= (C) Optional parameter to define the transform field quantity (calculated via the **SHAP DATA** command) to plot. If not input, a deformed material plot is displayed. If a shape plot of a field quantity such as pressure is displayed with this **PLOT** command, the pressure data array that was computed to make the plot will have been saved in the data array LAST in the Data Manager SHAP directory. These data are then accessible to the user before other shape plots that may overwrite this array are produced.

phaseshift= (F) Phase shift [degrees] to apply to the shape before plotting. Default = 0.

<<FLXHST - option to allow plotting of current time histories being save in the PZFlex run>>

This option allows the user to plot the current state of time histories requested to be saved by the **POUT** command group.

irecordn= (I) Time history record numbers to plot. Note that the **LIST** command in **PZFlex** list the time history records. The plotting functionality is the same as if these time histories were read into Review. Therefore, up to six records can be cross-plotted simultaneously.

Rows of properly mapped data arrays may be plotted via the constructl araynam>n where n is the row number. Aliases have been defined for plotting quantities of frequent interest. These are listed here. For example, PLOT shel_ep11 displays the 11-strain component in shell elements.
Point quantities are displayed with a box around the point. Box size and aspect ratio may be adjusted via SET/PSET options.

ALIAS	Quantity
plod_pres	Plod pressure
plod_range	Plod range (for conwep)
shel_ep11	Shell strain component
shel_ep22	Shell strain component
shel_ep12	Shell strain component
shel_ep13	Shell strain component
shel_ep23	Shell strain component
shel_epls	Shell equivalent plastic strain
shel_sg11	Shell stress component
shel_sg22	Shell stress component
shel_sg12	Shell stress component
shel_sg13	Shell stress component
shel_sg23	Shell stress component
bm3d_ep11	Beam strain component
bm3d_ep22	Beam component
bm3d_ep12	Beam strain component
bm3d_ep13	Beam strain component
bm3d_ep23	Beam strain component
bm3d_epls	Beam equivalent plastic strain
bm3d_sg11	Beam stress component
bm3d_sg22	Beam stress component
bm3d_sg12	Beam stress component
bm3d_sg13	Beam stress component
bm3d_sg23	Beam stress component

ALIAS	Quantity
bar_ep	Bar axial strain
bar_sg	Bar axial stress
bar_epls	Bar equivalent plastic strain
rigd	Rigid substructures
rigd_xvel	Rigid substructure x velocity
rigd_yvel	Rigid substructure y velocity
rigd_zvel	Rigid substructure z velocity
rigd_rxvel	Rigid substructure x rotational velocity
rigd_ryvel	Rigid substructure y rotational velocity
rigd_rzvel	Rigid substructure z rotational velocity
pnbl	Pinballs
pnbl_rad	Pinball radii
pnbl_stiff	Pinball stiffness
pnbl_xfrc	Pinball x force
pnbl_yfrc	Pinball y force
pnbl_zfrc	Pinball z force
pnbl_pen	Pinball penetration
pnbl_xcrd	Pinball x coordinate
pnbl_ycrd	Pinball y coordinate
pnbl_zcrd	Pinball z coordinate
piez	Electrodes
piez_voltage	Electrode voltage
piez_charge	Electrode charge
piez_current	Electrode current
boun_name	Boundary condition names
boun_type	Boundary condition types
bond_coarse	Coarse side bonds
bond_fine	Fine side bonds
spring	Line elements
it3d>xx	Slideline data array value xx
it3s	Slideline spring values

Note: If RANG and *minvalue /maxvalue* is input, the color scale is set appropriately. If the data array being displayed contains data values that fall outside the specified display range they are assigned the nearest color, and the end points of the numerical scale plotted on the screen are adjusted accordingly. Numerical values associated with interior colors of the color scale are unaffected.

Examples:

<< plot material properties assigned to grid >>

```
PLOT  MATR                      /* plots material properties displayed on the model

PLOT  IJK MATR                  /* plots material properties displayed on the model in ijk-space
PLOT  MATR J 5                  /* plots material properties for the J=5 slice of the model
PLOT  MATR K 1 8 J 1 5 I 1 8        /* plots material properties for the subregion of the
                                       model indicated by the indices ranges shown
BLOK B1 1  8 1 5 1 8            /* defines a subregion, B1, of the model to plot
PLOT PRES BLOK B1               /* plots the pressure for the subregion defined by block B1.
                                   Same subregion as previous example
```

<< plot spatial distribution of field variables on the grid >>

```
PLOT  XVEL                      /* plots the x-velocity field for the entire model
PLOT  XVEL I 30 XVEL K 20       /* plots the I=30 and k=20 slices of the x-component of velocity
                                   field
PLOT PRES BLOK B1               /* plots the pressure for the subregion defined by block B1.
PLOT SGXX J 6 K 8 VS. X         /* plots the xx-normal-component of the stress field along a
                                   single column of elements having j-index = 6, k-index = 8,
                                   and default i-indices ranging from 1 to the maximum i-
                                   index as an xy plot of value verses x-coordinate.
PLOT ARR1 I 1 100 J 6 6 K 8 8 VS ARR2 I 1 1 J 1 100 K 8 8
                                /* plot a 1-D vector from ARR1 defined from i =1 to 100 vs
                                   a 1-D vector from ARR2 defined from j=1 to 100
```

<< plot extracted resonance shapes at **SHAP** specified frequencies >>

```
PLOT SHAP 2 90.                 /* plots the deformed shape (showing materials) of the 2nd
                                   frequency requested by the SHAP command group with a
                                   90-degree phase shift

DISP 2.                         /* sets the deformation scale factor to 2.
PLOT SHAP 3 PRES 180            /* plots the field distribution of the pressure field at the 3rd
                                   frequency requested by the SHAP command group with a
                                   180-degree phase shift. User must have requested that
                                   shape information for pressure be computed.
```

<< plot time histories being save in this run. Requested by **POUT** command >>

```
PLOT FLXHST 1                   /* plots the first time history record being saved during this
                                   computation.
PLOT 1                          /* shorthand way to plot the first record
PLOT FLXHST 1 3 5               /* cross-plots records 1, 3, and 5
```

PS (PostScript) Subcommand
GRPH (graphics) Input Group **GRPH-PS**

Purpose: To make PostScript image files for display on PostScript-compatible devices such as laser printers. A graytone PostScript image file is produced by default. To produce a color PostScript image file, the **SET** COLR ON subcommand must have been input.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Must precede and/or follow the graphics commands that generate the image.

PCOM	SCOM	input parameters
PS	*option*	

option = (C) Set to NOW, FILE, ON or OFF. If set to NOW, creates a PostScript image of the current graphics display. If set to ON, creates a single PostScript image of any following PLOT commands until a PS OFF subcommand is encountered.
Default = NOW, unless a PS ON subcommand was input prior, in which case default = OFF.
If set to FILE, a new root file name can be entered for a following set of images. The page numbering is reset to 1.

Note: Any number of PostScript images may be created. All are contained on the PostScript output file. The entire file may be routed to a PostScript-compatible printer for plotting. The **PS** command group of **Review** allows for splitting a multi-image PostScript file into individual image files. Each page is a separate PostScript file numbered sequentially.

Example:
```
GRPH
   NVEW 1        /* ses number of views to 1
   PS ON /* initiates PostScript plotting for this image
   PLOT DAT1     /* plots DAT1 in view 1
   PS OFF /* terminates PostScript plotting for this image
   NVEW 2        /* sets number of views to 2
   PLOT DAT1     /* plots DAT1 in view 1
   PLOT DAT2     /* plots DAT2 in view 2
   PS       /* creates PostScript image of the current display
   END
```

REVR (reverse) Subcommand
GRPH (graphics) Input Group **GRPH-REVR**

Purpose: Reverse the axis specified for plotting for the next **PLOT** command.

Use Is: Optional.

Multiple Use: Allowed.

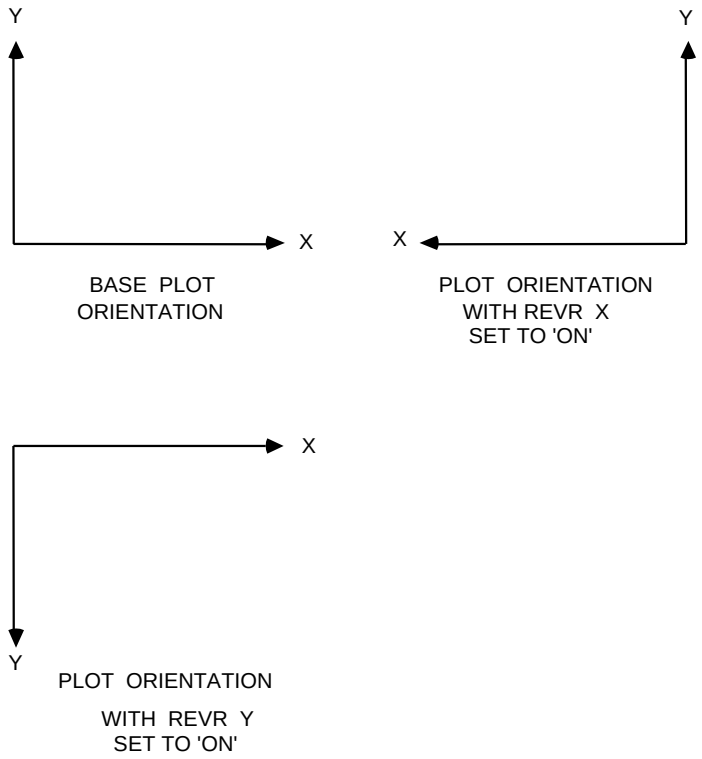
Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

REVR *iaxis option*

iaxis = (C) Axis to reverse.
Any of: X, Y or Z
No default.

option = (C) Option. Either ON or OFF.
Default = ON.



Effect of **REVR** subcommand on view orientation for 2D models

Note: For 3D models the view orientation is set with the **EYE**, **VPNT**, and **VERT** subcommands.

SET/PSET (set/preset) Subcommand
GRPH (graphics) Input Group **GRPH-SET/PSET**

Purpose: To set various graphics parameters. Syntax for the **SET** and **PSET** subcommands is the same. If applicable, the **SET** option sets graphics parameters for the current view. The **PSET** subcommand changes the code default for the parameter for all future plots. For options that cannot be changed once the plot is drawn, use of the **SET** command is interpreted as the **PSET** command.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: The action of the **PSET** subcommand applies to all following plots. The action of the **SET** command applies only to the current plot.

PCOM	SCOM	input parameters
	PSET	*option parameters*
	SET	*option parameters*

option = (C) The option to set with this subcommand.

parameters = Number and types of input parameters vary depending on *option* as described below.

for option = TERM; lets the user turn off the graphics display on a terminal to allow PostScript graphics images to be created in batch mode and on systems that do not support X11 graphics display.

PSET TERM *switch*

option = (C) Set *option* to X or OFF. Default is X. Use this option as the first input subcommand after the first occurrence of the **GRPH** command in order to deactivate the graphics terminal. If omitted or set to X, X11 graphics libraries must be present. If set to OFF, no graphics operations to the terminal screen are performed. This allows the generation of PostScript graphics in batch mode.

for option = FORM; allows the generation of PostScript images in either landscape or portrait form.

PSET FORM *type*

type = (C) Set *type* to LAND (landscape) or PORT (portrait). Default is LAND. This option affects only the PostScript image that is generated. It does not affect the screen image.

for option = WNDO; causes a time history or x-y plot to be displayed using the horizontal and vertical data ranges provided by the user. The **PSET** option applies to all future plots. The **SET** option applies to the current active view (either the last view displayed or the view set with the **VIEW** subcommand) unless the *alloption* parameter is specified. The window can be deactivated with the OFF option.

PSET/SET WNDO *xbegin xend ybegin yend alloption*
or
PSET/SET WNDO *option*

xbegin, xend = (F) Beginning and ending range of the horizontal data window that the user wants to display. Typically, this is the time range for the x-y plot. Default is previous values.

ybegin, yend = (F) Beginning and ending range of the vertical data window that the user wants to display. Typically, this is the data range for the x-y plot. Default is previous values.

alloption = (C) If set to ALL, the data window specified is set for all views currently displayed.

option = (C) Set to OFF to deactivate window specification

for option = DATE; controls whether or not the creation data and time are placed on the PostScript image.

PSET DATE *switch*

switch = (C) Set *switch* to ON or OFF. Default is ON. If ON, creation date and time are placed on the Postscript image.

for option = LW; controls the line width used to draw a PostScript image. The default line width is .75 points. This option applies to all views.

PSET LW *width*

width = (F) Width of lines in points where 1 point = 1/72 inch

for option = LC; controls the line color of curves drawn on an x-y plot. If a curve is given a line color, it is also given a line quality of 1 (solid line). A user may assign an alternative line quality after the line color has been assigned.

PSET/SET LC *icurve icolor*
or
PSET/SET LC *option*

icurve = (I) Curve number (between 1 and 6)

icolor = (I) Color number to assign to *icurve*

option = (I) Set to ON for all curves to have default colors 1 thru 6. Set to OFF to turn line colors off for all curves.

for option = LQ; controls the line quality to use for plotting x-y curves.

PSET/SET LQ *icurve linequal alloption*

icurve = (I) Curve number whose line quality is to be changed. This allows control over the line quality of each curve in a cross-plotted x-y view.

linequal = (I) Line quality to use (1= solid, 2= dashed, etc.). If positive value from 1 to 6, line quality is set to the specified value. If 0, line disappears from view. If negative value from -1 to -6, line quality is set to the absolute value of *linequal* and the curve legend disappears from the view.

alloption = (C) If set to ALL, the line quality specified is set for all x-y type views currently displayed.

for option = XLAB; change the label of the horizontal axis for an x-y plot

PSET/SET XLAB *label alloption*

label = (C) Label to use, may be up to 80 characters. Enclose in single quotes (') if the label contains embedded blanks.

alloption = (C) If set to ALL, the x label specified is set for all views currently displayed.

for option = YLAB; changes the label of the vertical axis for an x-y plot

PSET/SET YLAB *label alloption*

label = (C) Label to use, may be up to 80 characters. Enclose in single quotes (') if the label contains embedded blanks.

alloption = (C) If set to ALL, the y label specified is set for all views currently displayed.

for option = CLAB; change the curve label

PSET/SET CLAB *icurve label alloption*

icurve = (I) Curve number of the label to be changed. This allows control over the labeling of each curve in a cross-plotted x-y view.

label = (C) Label to use, may be up to 80 characters. Enclose in single quotes (') if *label* contains embedded blanks.
Note: Setting the last character of *label* to the plus (+) character causes *label* to be used in addition to (and preceding) the current or default curve label.

alloption = (C) If set to ALL, the curve label specified is set for the *icurve* curve for all views currently displayed.

for option = SWAP; swap the x and y axes for x-y plots

SET SWAP *switch*

switch = (C) Set *switch* to ON or OFF. Default is OFF. If ON, the x and y axes are swapped for the current view.

for option = COLR; requests that color PostScript files be generated instead of graytone ones when using the **PS** subcommand. Color PostScript image files are slightly larger than graytone files. Color files may be printed on any PostScript-compatible device. If the device does not support color, the image is converted to a graytone-equivalent image.

PSET/SET COLR *switch*

switch = (C) Set *switch* to ON or OFF. Default is OFF. If ON, color PostScript files are generated.

for option = STTL; defines a subtitle to be placed at the bottom of a view window the next time the view is drawn on the screen.

PSET/SET STTL *iview subtitle*

iview = (I) View number for which this subtitle is to be used.

subtitle = (C) Subtitle to use, may be up to 80 characters. Enclose in single quotes (') if the subtitle contains embedded blanks.

for option = BAR; control whether a color bar is placed on field plots and how the numeric color bar marker values are displayed

PSET/SET BAR *option*

option = (C) any of: STND = for linear data arrays (default)
LOG = displays actual values instead of log values for log data arrays
LN = displays actual values instead of log values for natural log data arrays
OFF = bar is not plotted

for option = BACK; for PostScript images only, controls background shading for graphics image. Default is white; if background option is set to ON, background shading is light gray.

PSET/SET BACK *switch*

switch = (C) either ON or OFF.

for option = LOG; to convert x-y graph scales from linear to logarithmic. Applies to the x-y graph in the current view.

PSET/SET LOG *axis switch*

axis = (C) either X or Y.
switch = (C) either ON or OFF.

for option = AXIS; to modify the position of the axis orientation figure for succeeding plots.

PSET/SET AXIS *option*

option = (C) any of: TL = top left of image (default)
TR = top right of image
BL = bottom left of image
BR = bottom right of image
OFF = no axis displayed

for option = TRI; to produce tripartite graphs. Automatically sets log scales for x and y axes. X-axis must be frequency. Used for plotting response spectra.

SET TRI *option scaleacc labelacc scaledisp labeldisp*

option = (C) either ON or OFF. Default=ON.

scaleacc, scaledisp = (F) Scale factor to convert to desired acceleration and displacement units. Default=1.

labelacc, labeldisp = (C) 4-character labels for accelerations and displacements, i.e., g, in. Default: *labelacc*=acc, *labeldisp*=disp

for option = MATR; redefines the label used to identify a material when plotting the model with the **PLOT** MATR subcommand. The label is displayed next to the color bar. If not input, the original material property set name is used.

PSET/SET MATR *matname matlabel*

matname = (C) the name of a material property set

matlabel = (C) the label to display identifying this material. May be up to 12 characters long.

for option = IMAG; selects the form of the image file written by all following **IMAG** subcommands.

SET IMAG *imagoption*

imagoption = (C) Form of the image file to write when an **IMAG** subcommand is input. Any of

STND = Standard form, a single file contain all images and the file is readable by the Review's **FILM** command group. Typically used to create movies.

TIFF = Writes an individual image file in TIFF format each time an **IMAG** subcommand is input. Display or processing of this file requires the use of other software that supports the TIFF standard.

XWD = Writes an individual image file in XWD format each time an **IMAG** subcommand is input. Display or processing of this file requires the use of other software that supports the XWD standard.

AVI = This form is available only on systems running a Microsoft Windows Operating system. A single file consisting of all images saved with each IMAG subcommand is created.

Imagoption_parameters = (X) Parameters depend on *imageoption.*

For imagoption = AVI

SET IMAG AVI (*frames_per_second*) *codec (avifile)*

frames_per_second = (I) Optional parameter. Specifies the number of frames per second for animation playback. Each **IMAG** subcommand creates a single frame of the animation. If not input, 15 frames per second is assumed.

Codec = (C) Either CVID or MENU. If set to CVID, the cinepak codec (Quicktime Compact Video by Radius), available on most Windows systems is used. If set to MENU, the first IMAG subcommand produces a pop-up graphical window listing all encoding options available on the system. The user is required to select the codec to use from the menu provided before the program will continue. *NOTE: Users should be aware of any color depth and/or display window size constraints on the various codecs before changing from the default setting.*
Default = CVID

avifile= (C) Optional parameter. Avi file name.
Default =*jobname.avi*

for option = POV; activates or deactivates the creation of POV (Persistence of Vision)-compatible geometry files when plotting a 3D model. If set to ON each time the model is plotted, the geometry is written to the flxpov.*name.n*.

SET POV *switch*

switch = (C) either ON or OFF.

for option = DB; to convert x-y graph scales from linear to dB scale according to the relation:

$$f'(x) = 20\log 10\left(\frac{f(x)}{dbvalue}\right).$$

PSET/SET DB *axis switch dbvalue*

axis = (C) either X or Y.

switch = (C) either ON or OFF.

dbvalue = (F) Value to use for computing dB. If not input, the max value for the curve in the current window is used.

for option BOXSIZE; Sets box size used for displaying nodal data groups such as pinballs, electrode nodes, and rigid substructures.

SET BOXSIZE *size*

size = (F) Size of cube. Default computed by code.

for option BOXSCALE; Set scale factors for box used in displaying nodal data groups such as pinballs, electrode nodes, and rigid substructures.

SET BOXSCALE *xscale yscale zscale*

xscale, yscale, zscale = (F) Scale factors to change cube into a box. Default = 1.0.

for option FILL; Set whether plots of a model's materials are shown with color fill representing each element's material property name or without color.

SET FILL *fill_option*

fill_option = (C) Set to either ON or OFF. If OFF, then the model is plotted without color. Default = ON.

for option BM3D; Set integration point number for which quantity is to be displayed.

SET BM3D *integration_point*

integration_point = (I) Integration point to display. Default = 1.

for option SHEL; Set layer number for which quantity is to be displayed and control whether shell thickness plotting is enabled. Note that for the default *num_option,* layer 1 means the layer closest to the viewer, 2 next farthest away, etc. Shell thickness plotting is inactive by default.

SET SHEL *layer thick num_option*

or

SET SHEL *option thick*

layer = (I) layer to display. Default = 1.

option = (C) Any of MIN, MAX, AVG, MAXABS. To select the minimum, maximum, average of maximum absolute value respectively.

thick = (C) Set to either ON or OFF. If OFF, then shells are plotted without thickness. Default = OFF.

num_option = (C) Set to either RELATIVE or ACTUAL. If RELATIVE, then shell layer numbers are interpreted relative to the viewpoint. If ACTUAL, then actual shell layer numbers are used for plotting. Default = RELATIVE.

for option = NPOINTS; controls number of x-y points to be plotted.

PSET/SET NPOINTS *npoints icurve*

npoints = (I) the number of points to be plotted, the first *npoints* points are plotted.

icurve = (I) The curve number for which NPOINTS are set.

for option NTIC: Sets the number of intervals in each coordinate direction for mapped arrays.

SET/PSET NTIC *nitic njtic nktic alloption*

nitic, njtic, nktic = (I) number of intervals in each coordinate direction. Default = 2

alloption = (C) if set to ALL, the intervals are set for all views currently displayed.

for option SPRING; Sets line element (spring) material or model type to be displayed . All line elements are displayed by default. OFF option turns off previous settings.

SET SPRING NAME *materialname*
or
SET SPRING TYPE *modeltype (column)*
or
SET SPRING OFF
or
SET SPRING BROKEN *brokenoption*

materialname = (C) Line element material to display. No default.

modeltype = (C) Line element model type to display. No default.

column = (C) For line elements types BOLT and JOINT only.
Set to X, Y, or Z to indicate column in LNRD for plotting. See table under LINE command for values stored in each column.
Default = X.

for option BUFMODE; Set graphics buffer mode.

SET/PSET BUFMODE *option*

option = (C) ON or OFF, Default = ON.
Currently implemented only for Windows, Not Linux.
OFF draws all graphics to the display, ON draws only the final bitmap to the display. ON is much faster, especially for remote access.

for option BLOK; Set graphics blok option.

SET/PSET BLOK *option*

option = (C) Set to TOUCH or ALL, Default = ALL.
If set to TOUCH then any general connectivity element (e.g. beams, bars, shells, lines, hexes) that is touching a blok range (i.e. has at least one node in) is plotted. If set to ALL then an element must be entirely within the blok range (i.e. all of the element nodes must be in) to be plotted.

for option XFORMAT; Change x-axis numeric format for line plots.

SET/PSET XFORMAT *option*

option = (C) Set to a valid FORTRAN format. E.g. '(1p,40e10.3)'.

for option YFORMAT; Change y-axis numeric format for line plots.

SET/PSET YFORMAT *option*

option = (C) Set to a valid FORTRAN format. E.g. '(1p,40e10.3)'.

for option ACE; Set graphics option for viewing ACE surfaces.

SET/PSET ACE *option*

option = (C) Set to either TOWARDS, AWAY or BOTH, Default = BOTH.
If set to TOWARDS only ACE blast surfaces facing the blast are plotted, if set to AWAY only ACE surfaces facing away from the blast are plotted, if set to BOTH all surfaces are plotted.

for option = FORMAT; controls the number format to use for plotting labels.

PSET/SET FORMAT *object format alloption*

object = (C) the object type to which the format description applies. Any of:
X (x-axis labels for x-y plots)
Y (y-axis labels for x-y plots)
BAR (color bar labels for field plots)
TIME (time label for field plots)
SCALE (scale label for field plots)

format = (C) a format descriptor for floating point numbers using fortran convention. Common formats are:

F*w.d* (real values, no exponent)
E*w.d*[E*e*] (real values, with exponent)
ES*w.d*[E*e*] (scientific notation)

where, *w* = max field width, *d* = number of digits after the decimal, *e* = number of digits in exponent

e.g. f7.1 prints 1234.567 as 1234.5
e12.5 prints 1234.567 as 0.12345E+04
es12.5 prints 1234.567 as 1.23456E+03

Default varies by object. Set *format* = OFF to revert to default.

alloption = (C) if set to ALL, then the format specified is set for all views currently displayed.

for option = VIS; controls visibility of plot labels.

PSET/SET VIS *object toggle alloption*

object = (C) the field plot object type to which the visibility applies.
Any of: TIME (time label)
STEP (timestep label)
SCALE (scale label)
BARLABEL (bar label)
ALL (all of the above)

toggle = (C) either ON or OFF to turn on/ off visibility.

alloption = (C) if set to ALL, then the visibility is set for all views currently displayed.

SIZE Subcommand
GRPH (graphics) Input Group **GRPH-SIZE**

Purpose: To allow the user to make PostScript x-y plots with a specified scale/inch This option does not affect the size of the image on a graphics terminal.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	SIZE	*xsize ysize xbegin xend ybegin yend nxdiv nydiv*

xsize = (F) Horizontal dimension of the graph window.
Set to OFF to deactivate preceding **SIZE** subcommand.

ysize = (F) Vertical dimension of the graph window.

xbegin, xend = (F) Beginning and ending values defining the horizontal data range.

ybegin, yend = (F) Beginning and ending values defining the vertical data range.

nxdiv = (F) Number of horizontal divisions on graph.

nydiv = (F) Number of vertical divisions on graph

Notes: 1. The subcommand:
SIZE OFF
causes future PostScript plots to be made using the code's auto-scaling logic.
2. Only one set of size specifications can be specified at any one time. All views that are being displayed will be affected by the **SIZE** subcommand.

SWAP Subcommand
GRPH (graphics) Input Group

GRPH-SWAP
(2D models only)

Purpose: Swap the plotting axes for 2D models. Horizontal becomes vertical, vertical becomes horizontal for the next **PLOT** command.

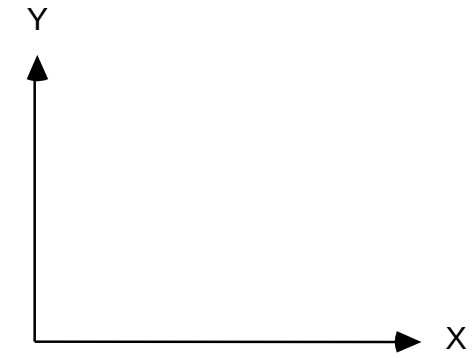
Use Is: Optional.

Multiple Use: Allowed.

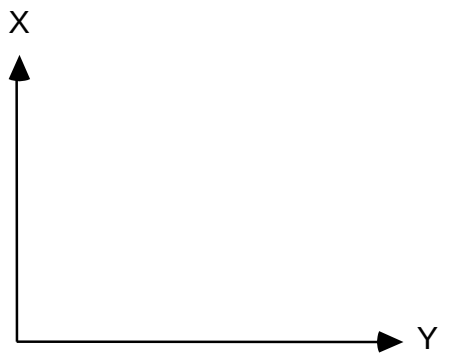
Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	SWAP	*option*

option = (C) Option. Either ON or OFF.
Default = ON.



BASE PLOT ORIENTATION
(SWAP = OFF)



PLOT ORIENTATION AFTER SWAP
(SWAP = ON)

SWAP Subcommand
GRPH (graphics) Input Group

GRPH-SWAP
(2D models only)

Purpose: Swap the plotting axes for 2D models. Horizontal becomes vertical, vertical becomes horizontal for the next **PLOT** command.

Use Is: Optional.

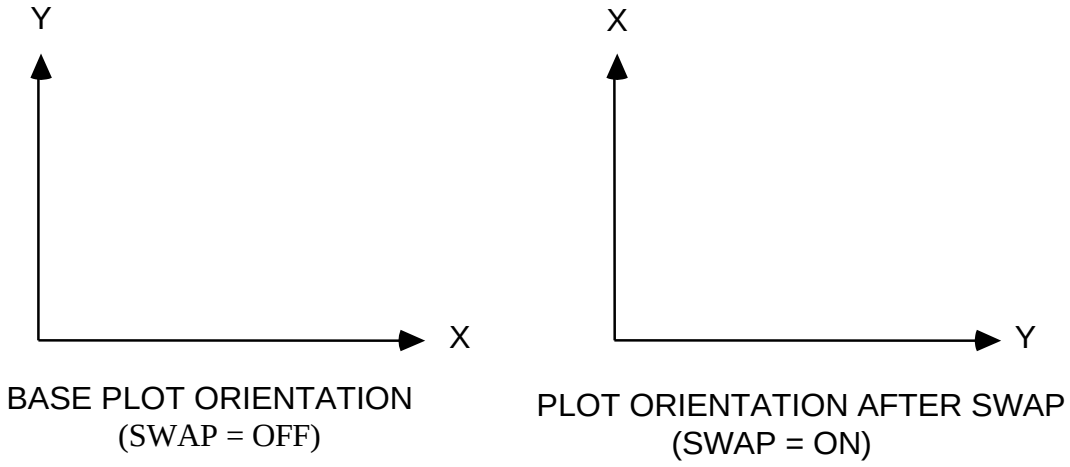
Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM — input parameters

SWAP *option*

option = (C) Option. Either ON or OFF.
Default = ON.



BASE PLOT ORIENTATION
(SWAP = OFF)

PLOT ORIENTATION AFTER SWAP
(SWAP = ON)

TIME Subcommand
GRPH (graphics) Input Group **GRPH-TIME**

Purpose: To control the values of time and timestep number that appear on the next graphic image to be displayed. This command is used only if the user desires to change the time parameters from those of the current execution time of the model. Typically, this might be done when doing postprocessor to display snapshot results generated at various times during a preceding analysis.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Affects the time information displayed by each succeeding **PLOT** subcommand.

PCOM SCOM input parameters

TIME *option* (*itimestep time*)

option = (C) Graphics time information option.
May be any of: SET, DATA, EXEC

If DATA - the values for timestep number and time in succeeding graphics plots are taken from the time and timestep number of the last data array imported into the program using a **DATA IN** command.

If EXEC- the values for timestep number and time in succeeding graphics plots are taken from the current execution time of the model.

If SET - the values for timestep number and time in succeeding graphics plots are taken from the *itimestep* and *time* input parameters.

No default.

itimestep = (I) Timestep number to display if *option* = SET.
Default = 0

time = (F) Time to display if *option* = SET.
Default = 0.0

TTL (title) Subcommand
GRPH (graphics) Input Group **GRPH-TTL**

Purpose: To provide titling information for the graphic currently displayed on the screen. The title is replaced without redrawing the graphic.

Use Is: Optional. If no **TTL** subcommand is input, the title of all graphics is set to the title taken from the **TITL** command.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

TTL *ititle*
<----------200 character title information for line = *ititle*---------->

ititle = (I) Title line number.
The next input line (starting with column 1) contains up to 200 characters of text to substitute for title line = *ititle.* Up to three lines of titling information may be provided. Blank title lines are allowed if no title is desired.
(i.e., $1 \leq ititle \leq 3$).

Default = 1

Note: Once specified, the title information is used for all succeeding graphics until changed with the **TTL** subcommand.

VERT (vertical point) Subcommand
GRPH (graphics) Input Group

GRPH-VERT
(3D models only)

Purpose: To define the vertical point used to define the view orientation of the following plots. The other components of the viewing orientation are provided by the **EYE** and **VPNT** subcommands.

Use Is: Optional. Default is x=-1., y=0.0 and z=0.0.

Multiple Use: Allowed.

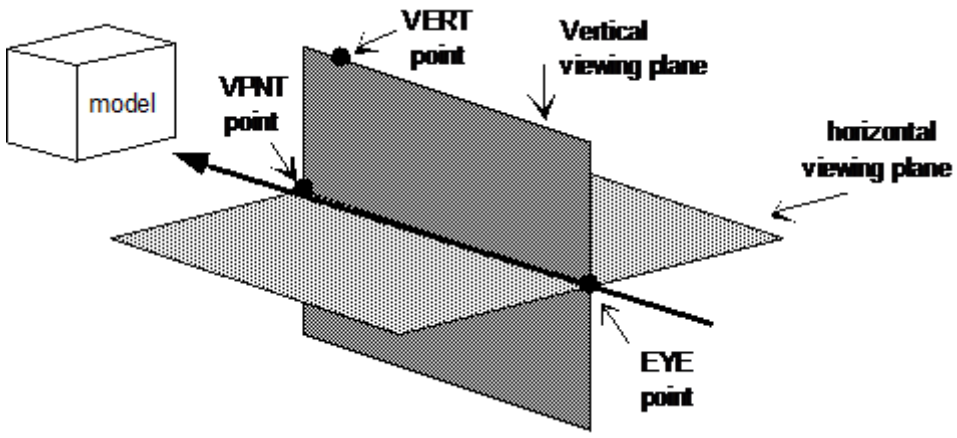
Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	VERT	*xvalue yvalue zvalue*

xvalue = (F) x-coordinate of the vertical point.
Default = 0.0.

yvalue = (F) y-coordinate of the vertical point.
Default = 0.0.

zvalue = (F) z-coordinate of the vertical point.
Default = 0.0.



Definition of viewing orientation for 3D models

VIEW Subcommand
GRPH (graphics) Input Group **GRPH-VIEW**

Purpose: Specify the next view window to plot within. View numbers are as indicated for the **NVEW** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	VIEW	*iview*

iview = (I) Viewing window to use for the next plot .
Applicable only for *nview* > 1.
Default = 1

Note: If the **VIEW** subcommand is not input, the plot window cycles from 1 through *nview* and then back to 1 as each **PLOT** subcommand is encountered.

VPNT (view point) Subcommand
GRPH (graphics) Input Group

GRPH-VPNT
(3D models only)

Purpose: To define the view point used to define the view orientation of the following plots of 3D models. The other components of the viewing orientation are provided by the **EYE** and **VERT** subcommands.

Use Is: Optional. Default is x=0.0, y=0.0 and z=0.0.

Multiple Use: Allowed.

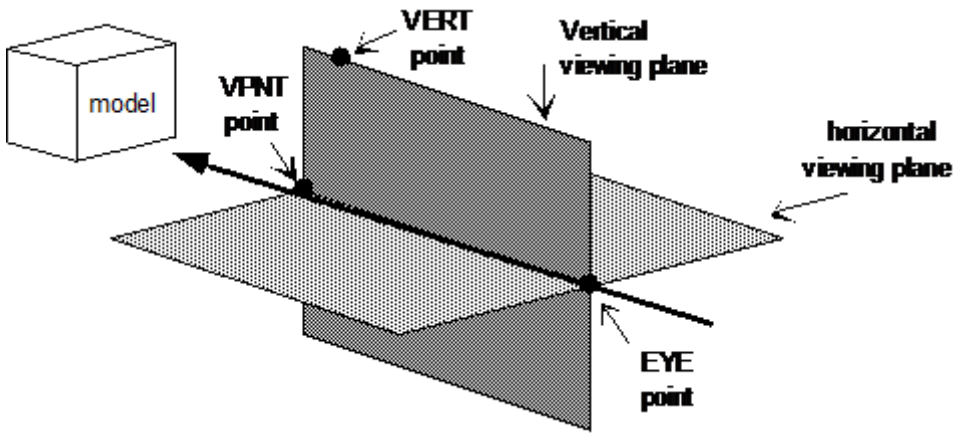
Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	VPNT	*xvalue yvalue zvalue*

xvalue = (F) x-coordinate of the view point.
Default = 0.0.

yvalue = (F) y-coordinate of the view point.
Default = 0.0.

zvalue = (F) z-coordinate of the view point.
Default = 0.0.



Definition of viewing orientation for 3D models

WNDO (window) Subcommand
GRPH (graphics) Input Group **GRPH-WNDO**

Purpose: Specify the size of the plotting window in model units; otherwise model is sized to fit the viewport.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	WNDO	*(option1) (option2) xbegin xend ybegin yend zbegin zend*

Option1 = (C) Optional input parameter. If set to ELEM, elements that fall within the spatial window are plotted. Otherwise, the spatial window defines the "field of view" but does not exclude elements that fall outside the window from being displayed.

option2 = (C) Optional input parameter. Must follow *option1* if *option1* is entered. If set to CURR, then the current window coordinates are stored as the bounds for subsequent plotting windows. The following coordinates are not entered if this option is set.

xbegin, xend = (F) Beginning and ending x-coordinate of window
No default.

ybegin, yend = (F) Beginning and ending y-coordinate of window
No default.

zbegin, zend = (F) Beginning and ending z-coordinate of window. For 3D models only.
No default.

Note: To deactivate the window once it is set, the **WNDO** OFF subcommand must be input.

WRIT (write) Subcommand
GRPH (graphics) Input Group **GRPH-WRIT**

Purpose: To write out the x and y values of xy plots in the current view to a file as column data.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

WRIT (*options*) *filename*

options = (C) Optional input parameters. May be set to any or all of: ONEX, NOLB, FORM, APPEND. If more than one curve is plotted in the current view, set to ONEX to have only the x-data values from the first curve written out with the y-data values for all curves. Otherwise, the x-data values from each curve are written in column form. If curve labels for each column of data are not desired, they may be suppressed with the NOLB (no label) option. The FORM option allows the user to change the default output format of the data. This option must be followed by a format specification. See Note 4 below. The APPEND option appends data to an existing file.

filename = (I) File name to which the column data are to be written.

Notes: 1. The x-y data written to *filename* is for the current view, which is the last view plotted. The current view may be changed by using the **VIEW** subcommand.
2. The data written to *filename* is in column form. The first column contains the x-values of curve 1. The second column contains the y-values of curve 1. The third column contains the x values of curve 2, etc. This assumes that the ONEX option was not specified.
3. The first line of *filename* contains character labels identifying the data in each column unless the NOLB option was specified.
4. Up to 12 columns may be written. The default FORTRAN format is (1p,12e12.4). The user may use the FORM option followed by a format to override the default format. For example: WRIT FORM (12f15.3) MYFILENAME

WSIZ (window size) Subcommand
GRPH (graphics) Input Group **GRPH-WSIZ**

Purpose: To set the size of the graphics window on the terminal screen by command instead of manually with a mouse.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	WSIZ	*width height*

width = (I) Width of the window in pixels.

height = (I) Height of the window in pixels.

ZOOM (image zoom) Subcommand
GRPH (graphics) Input Group **GRPH-ZOOM**

Purpose: Zoom the "virtual camera" displaying the model "in" (move closer to the image) or "out" (move farther away from the image). This option in combination with the **PAN** subcommand allows for closer inspection of regions of the model currently displayed. The option applies only to images of the model or snapshots of field variables.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

ZOOM *zoom_value (all_option) (save_option)*

zoom_value = (F) Zoom magnification. If greater than 1, the image appears closer. If less than 1, the image appears farther away. A zoom_value of 2, makes the image appear twice as close to the viewer.
Default = 1.0, this equivalences to the original plot display.

option = (C) If set to ALL, all current views that contain field plots are redrawn with the zoomed magnification. If omitted, this **ZOOM** subcommand applies only to the current view.
No default.

save_option = (C) If set to SAVE, then the zoom setting will be retained for the current view. Can be combined with the ALL option. Default = OFF — settings are cleared.

Note: When an image is displayed for the first time, a clipping border is established that bounds the image on the screen. The **ZOOM** subcommand does not alter this clipping border.

GSMU (Grid Smoothing) Input Command

Purpose: To smooth the material interfaces of a skewed window.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow **GRID & SITE** commands and precede **PRCS** command.

PCOM	SCOM	input parameters
GSMU		
	WNDO *ibeg iend jbeg jend kbeg kend*	
	PARM *bandwidth niter*	
	LIMIT *tlimit*	
	CALC	
	END	

Notes: 1. See Taubin, G., "Geometric Signal Processing on Polygonal Meshes," Eurographics, 2000.

WNDO (Window) Subcommand
GSMU Input Group **GSMU-WNDO**

Purpose: To define the mesh region to be smoothed. Default is entire skew partition.

Use Is: Not Required.

Multiple Use: Allowed.

Order Dependence
Within **GSMU** Group:Must precede the CALC command for this WNDO

PCOM	SCOM	input parameters
	WNDO	*ibeg iend jbeg jend kbeg kend*

ibeg, iend = (I) Window range in I direction

jbeg, jend = (I) Window range in J direction

kbeg, kend = (I) Window range in K direction

PARM (Parameters) Subcommand
GSMU Input Group **GSMU-PARM**

Purpose: To define the smoothing parameters.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
Within **GSMU** Group: Must occur before **CALC.**

PCOM	SCOM	input parameters
	PARM	*bandwidth niters*

bandwidth = (F) Typically values from 0.01 to 0.1

niters = (I) number of smoothing iterations

LIMIT Subcommand
GSMU Input Group **GSMU-LIMIT**

Purpose: To stop smoothing when its effect on timestep becomes too large.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GSMU** Group: Must occur before **CALC.**

PCOM	SCOM	input parameters
	LIMIT	*tlimit*

tlimit = (F) Permissible fractional reduction of timestep in the smoothing window (0 < tlimit < 1). 0.7 means that timesteps could be reduced to 70% of the unsmoothed value.

CALC Subcommand
GSMU Input Group **GSMU-CALC**

Purpose: To perform smoothing in the last defined window using the last defined parameters..

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
Within **GSMU** Group: Must follow applicable WNDO, PARM subcommands.

PCOM	SCOM	input parameters
	CALC	

HEAT Input Command

Purpose: To specify computational parameters specific to the thermal solver.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must precede **PRCS** command.

PCOM SCOM input parameters

HEAT

WNDO	*ibeg iend jbeg jend kbeg kend*
SLVR	*solvertype alpha droptol*
NLIN	*nstprfr maxits nitrfr rtol relax prnoption*
CUPL	*option*
END	

WNDO (window) Subcommand
HEAT Input Group

Purpose: To define the thermal window.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None.

PCOM SCOM input parameters

WNDO *ibeg iend jbeg jend kbeg kend*

ibeg, iend = (I) Beginning and ending nodal indices in i direction defining region of grid in which thermal solution occurs. Default: *ibeg=1*, *iend = indgrd*.

jbeg, jend = (I) Beginning and ending nodal indices in j direction defining region of grid in which thermal solution occurs. Default: *jbeg=1*, *jend = jndgrd*.

kbeg, kend = (I) Beginning and ending nodal indices in k direction defining region of grid in which thermal solution occurs. Default: *kbeg=1*, *kend = kndgrd*.

SLVR (solver) Subcommand
HEAT Input Group **HEAT-SLVR**

Purpose: To define the time integration scheme for thermal solutions.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM SCOM input parameters

SLVR *slvrtype alpha droptol*

slvrtype (C) Solver type. Any of: DRCT, DISK, CGDSor SPLU. Default = DRCT.

alpha (F) Time integration parameter for generalized trapezoid rule. Default = 0.5

droptol (F) Drop tolerance for SPLU solver. (See Note 3.) Default = 1.e-12. Alternatively, the relative error tolerance for the CGDS solver. Default = 10* machine precision (approx 1.e-6 in single precision). (See Note 4.)

Notes: 1. The DRCT option implements an in-core solver. This is the preferred solver if the problem is small enough to fit in available RAM. The DISK option implements an out-of-core direct solver. This is somewhat slower than the in-core solver, but is limited only by available disk, not RAM. CGDS is an iterative method using conjugate gradients with diagonal scaling. It is the most efficient method in terms of RAM requirements and does not require matrix factorization.. SPLU is a sparse direct solver.

2. For $\alpha > 0.5$, the time integration is unconditionally stable. Optimal accuracy is achieved at = 0.5, but stability characteristics improve as α is increased toward 1.0.
3. The SPLU solver tries to preserve sparsity during LU factorization. First the matrix is normalized to 1.0. During factorization, entries which become smaller than the drop tolerance are removed. Larger values reduce storage requirements, but reduce accuracy.
4. When the 2 norm of the residual divided by the 2 norm of the righthand side drops below *droptol*, the iteration terminates. Input values below default are set to default.

NLIN (nonlinear) Subcommand
HEAT Input Group **HEAT-NLIN**

Purpose: To set or reset parameters for a nonlinear thermal computation.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM SCOM input parameters

NLIN *nstprfr maxits nitrfr rtol relax prnoption*

nstprfe (I) Conductivity matrix is reformulated every *nstprfr* timesteps. Default = 1.

maxits (I) Maximum number of iterations per timestep. Default = 5.

nitrfr (I) Matrix is reformulated every *nitrfr* iterations. Default = 1.

rtol (F) Relative tolerance for convergence. Default = 1.e-5.

relax (F) Relaxation factor. Default = 1

prnoption (C) Print option. If set to PRNT, convergence information is printed to the flxprt file for each iteration. Default OFF

Notes:
1. This subcommand is used for nonlinear problems where the material properties are temperature dependent or the boundary conditions are nonlinear.
2. Direct iteration is used at each timestep to achieve the requested accuracy. Iteration at each timstep continues until $\|[A]T - Q\| \le rtol\left(\|[A]\|\|T\| + \|Q\|\right)$. Here [A] is the conductivity matrix, T is the temperature vector, and Q represents the forcing terms. At each iteration n, T is updated as: *relax* T^n + (1-*relax*)T^{n-1}. Values of relax less than 1 can be useful for strongly nonlinear problems.
3. Reformulation of the conductivity matrix is an expensive step. Mildly nonlinear problems can be solved more economically by reducing the frequency of reformulation.

CUPL (coupling) Subcommand
HEAT Input Group **HEAT-CUPL**

Purpose: To define field coupling.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM	SCOM	input parameters
	CUPL	*option (tref)*

option (C) Field coupling. Any of: OFF, MECH or LOSS

For option = OFF - No coupling is performed. Thermal analysis only.

For option = LOSS - Mechanical losses at each timestep are used as heat source terms for the thermal solver

For option = MECH — Mechanical stresses due to temperature changes are computed.

tref (F) Initial or reference temperature for MECH option. Default = 0.

Note: Typically, users perform decoupled analyses using the OFF option for thermal analyses or the MECH option for mechanical analyses. As the time scale related to temperature Buildup and thermal propagation within the model is usually orders of magnitude greater than that needed for mechanical wave propagation, the direct coupling between the mechanical wave field and thermal field is usually negligible.

JOB Input Command

Purpose: To specify the command processing option for job input files.

Use Is: Optional. If not input, defaults to batch option.

Multiple Use: Allowed.

Order Dependence: None.

PCOM input parameters

JOB *option*

option =	(C)	Any of:	BATC =	batch-style processing option
			INTR =	interactive-style processing option in memory.

Default = BATC

LDEF (large deformation) Input Command

Purpose: To define the skewed partition of the model to be a large deformation partition. This option accounts for the effects of large rotations occurring within the model. The option is available for 2D plane strain, 2D axisymmetric, and 3D models.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence:Must follow the **GEOM** command group and precede the **PRCS** command.

PCOM _____ input parameters _____

LDEF *option1 option2*

option1 = (C) The large deformation option to use in the analysis.
- Any of:
  - LDEF = update geometry to account for large deformation effects.
  - SDEF = do not update geometry; this equivalences to the small deformation approximation
- Default = LDEF

option2 = (C) The geometry update procedure to use if *option1* = LDEF
- Any of:
  - LPRC = update geometry procedures for low-precision computer systems
  - HPRC = update geometry using procedures for high-precision computer systems
- Default = LPRC for 32 bit systems.
  HPRC for 64 bit systems

Notes:
1. The LPRC option uses more memory than the HPRC option, as duplicate copies of the skewed coordinates are stored as well as the displacements for all nodes.
2. The large deformation option is fully implemented for most element types in the code. Any liner, line, or membrane elements still assume a small deformation approximation even though the LDEF command is input.
3. Boundary conditions are limited to the assumption that the geometry of the boundaries does not change during a calculation.
4. If this command is input, the stress data arrays are computed during the analysis.
5. This option requires that the B-matrix of each element be recomputed each time step. Therefore, any attempt to use the **BMAT** command to store instead of recompute the B-matrices is overridden.

LINE (line element) Input Command

Purpose: To define 2-noded spring and/or dashpot line elements for the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow **GRID** command and precede **PRCS** command.

PCOM	SCOM	input parameters
LINE		
	TYPE	*linetype*
	PROP	<--- input parameters vary depending on *linetype* -->
	CURV	*curvname (formoption) (hardoption) disp1 forc1 & disp2 forc2 ... dispn forcn*
	LDEF	*option*
	ELEM	*linename ibegin iend jbegin jend kbegin kend*
	ORFC	*n1 n2 area temperature gas_cons gamma*
	SNGL	*linename i1 j1 k1 i2 j2 k2 (i2 j2 k2 i3 j3 k3)*
	GCON	*linename (axisname) n1 n2 (n3 n4 area)*
	ITFC	*ib1 ie1 jb1 je1 kb1 ke1 ib2 ie2 jb2 je2 kb2 ke2 xref yref zref opt*
	PRNT	
	INFO	*option1 option2*
	END	

Notes:
1. The mass of line elements are lumped to the element's nodes.
2. The stability characteristics of line elements are considered when computing zone time steps. Warning! Currently, the stability time step computed for a line element containing a viscous component is not necessarily conservative. It is recommended that the element be tested using a simple checkout run to evaluate whether the element is stable for the zone time step assigned to the element. Also, a line element may not be attached to a massless node unless the time step override parameter *delt* is input.
3. Line element data are stored in the LNRD(i,j) data array. The same array is used for liner element data. The i index of this array ranges from 1 to 6. The model quantity indicated by the i-index depends on the type of line element as indicated in the table.

i-index	SPR1	SPR2	SPRG	VISC	MAXW	KELV	CABL	EOS	CAP	EACT	GSPR
1	axial force	axial force	axial force	axial force	axial force	axial force	axial force	axial force	axial force	Axial force	axial force
2	not used	not used	not used	not used	not used	not used	not used	pressure	axial stress	Voltage	alpha
3	axial relative motion	axial relative motion	axial relative motion	axial relative motion	axial relative motion	axial relative motion	axial relative motion	volume	lateral stress	Axial relative motion	relative disp for dof
4	not used	Xelastic	Xelastic	rate	not used	rate	Xelastic	energy	el	charge	uniax force
5	not used	not used	gap	not used	not used	not used	gap	volume 0	vs.ig	Not used	uniax displace
6	not used	not used	not used	not used	not used	not used	not used	mass	axial strain	Not used	axial strain

The j index ranges from 1 to the number of line elements in the order input.

i-index	JOINT X' DOF	JOINT Y' DOF	JOINT Z' DOF
1	axial force	Y' force	Z' force
2	Axial stress	Y' shear stress	Z' shear stress
3	axial relative motion	Y' relative motion	Z' relative motion
4	Peak axial stress (HYST)	Slip magnitude	dilation
5	DISP at peak stress	Not used	Not used
6	Last active segment	not used	not used

3 line elements are created for each JOINT spring called out

i-index	JOINT X' DOF	JOINT Y' DOF	JOINT Z' DOF
1	axial force	Y' force	Z' force
2	Axial stress	Y' shear stress	Z' shear stress
3	axial relative motion	Y' relative motion	Z' relative motion
4	Equiv. plastic strain	Not used	Not used
5	gfract	Not used	Not used
6	Not used	not used	not used

3 line elements are created for each BOLT spring called out

4. Line elements may be displayed with the GRPH PLOT SPRING command. The length of the spring element is the distance between the two nodes that make up the element. The width of the spring element can be controlled with the GRPH PSET BOXSIZE command. In 2D, line elements are plotted as lines. The GRPH SET SPRING command can be used to limit line element plotting to specific materials or model types.
5. The line element data stored in the LNRD array can be plotted using the extraction character ">" to indicate a specific model quantity. (e.g., LNRD>1 plots axial force)

TYPE Subcommand
LINE (line element) Input Group **LINE-TYPE**

Purpose: To define the type of line element model for the following **PROP** subcommands.

Use Is: Optional. If no **TYPE** subcommand is input, type SPR1 is assumed.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must precede all **PROP** subcommands which are of the type specified.

PCOM SCOM input parameters

TYPE *linetype*

linetype = (C) The type of line element model to assume for any following **PROP** subcommands.

Any of:

SPR1=	nonlinear elastic spring
SPR2=	linear elastic/plastic spring
SPRG=	linear elastic/plastic spring with gap
VISC=	nonlinear viscous element
MAXW=	Maxwell linear viscoelastic model
KELV=	Kelvin linear viscoelastic model
CABL=	nonlinear elastic/plastic cable element
EOS=	uniaxial strain eos model
CAP=	uniaxial strain cap model
GSPR=	general spring for any dof
EACT=	electostrictive actuator
JOINT=	frictional Interface
BOLT=	bolt spring

No default.

PROP (properties - SPR1) Subcommand
LINE lLine element) Input Group **LINE-PROP**

Purpose: To define model properties for a model with *linetype*=SPR1, a nonlinear elastic spring.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** SPR1 subcommand and precede any **ELEM** subcommand that references the defined line model.

PCOM SCOM input parameters

PROP *linename mass k1 k2 xmax delt*

linename = (C) Name of the line element model.
No default.

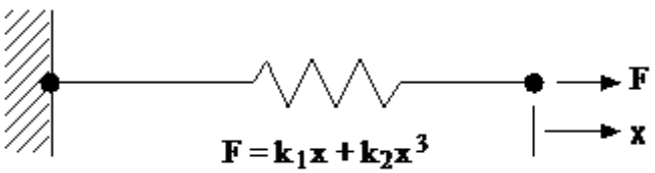
mass = (F) Total mass of the element.
No default.

k1 = (F) Value of spring constant k1.
No default.

k2 = (F) Value of spring constant k2.
No default.

xmax = (F) Conservative estimate of the maximum displacement that will occur during the simulation.
(used to compute stable time step for element).
No default.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model.
Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Spring element of *linetype* = SPR1

PROP (properties - SPR2) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=SPR2, a linear elastic/plastic spring.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** SPR2 subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass k fmax delt*

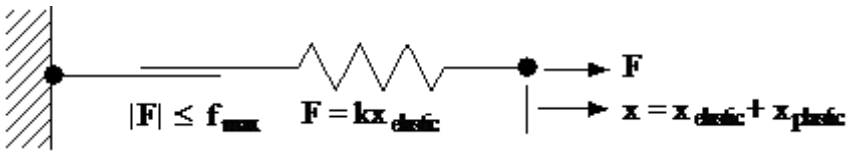
linename = (C) Name of the line element model.
No default.

mass = (F) Total mass of the element.
No default.

k = (F) Value of spring constant k.
No default.

fmax = (F) Peak compressive or tensile force the element may have.
Default: *fmax* = 0.0, no plastic limit, element is purely linear elastic.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model.
Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Spring element of *linetype* = SPR2

PROP (properties - SPRG) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=SPRG, a linear elastic/plastic spring with gap capability.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** SPRG subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass k fmax fcutoff gapinitial delt*

linename = (C) Name of the line element model. No default.

mass = (F) Total mass of the element. No default.

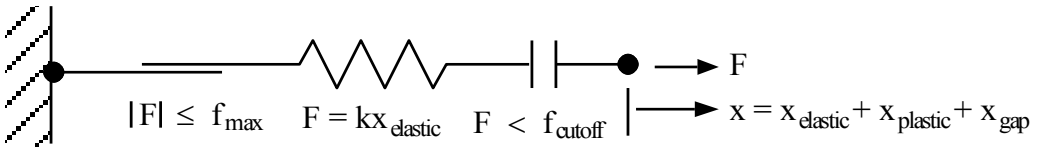
k = (F) Value of spring constant k. No default.

fmax = (F) Peak compressive or tensile force the element may have. Default: *fmax* = 0.0, no plastic limit, element is linear elastic with gap capability.

fcutoff= (F) Tensile force cutoff at which the gap begins to open. Default = 0.0

gapinitial= (F) Initial gap present in the element. Default = 0.0, no gap.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Spring element of *linetype* = SPRG

PROP (properties - VISC) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=VISC, a nonlinear viscous model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** VISC subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass c1 c2 vmax delt*

linename = (C) Name of the line element model. No default.

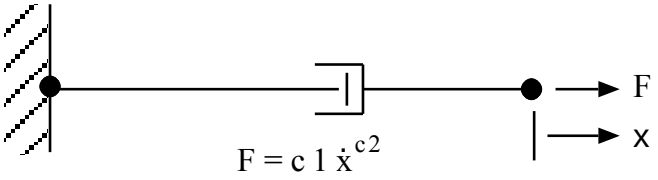
mass = (F) Total mass of the element. No default.

c1 = (F) Value of viscous parameter c1. No default.

c2 = (F) Value of viscous parameter c2. No default.

vmax= (F) Conservative estimate of the peak relative, longitudinal velocity of the two end nodes that will occur during the simulation (used to compute stable time step for element). No default

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Viscous element of *linetype* = VISC

PROP (properties - MAXW) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=MAXW, a linear Maxwell viscoelastic model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** MAXW subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass k c vmax delt*

linename = (C) Name of the line element model. No default.

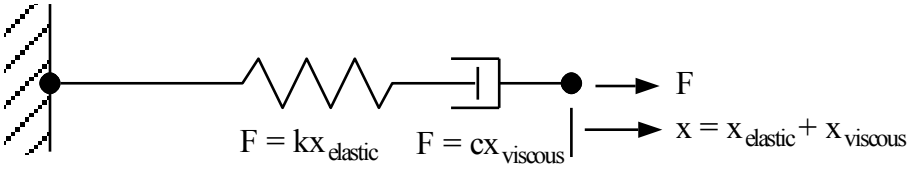
mass = (F) Total mass of the element. No default.

k = (F) Value of spring parameter k. No default.

c = (F) Value of viscous parameter c. No default.

vmax= (F) Conservative estimate of the peak relative, longitudinal velocity of the two end nodes that will occur during the simulation (used to compute stable time step for element). No default

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Viscous element of *linetype* = MAXW

PROP (properties - KELV) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=KELV, a linear Kelvin viscoelastic model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** KELV subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass k c vmax delt*

linename = (C) Name of the line element model. No default.

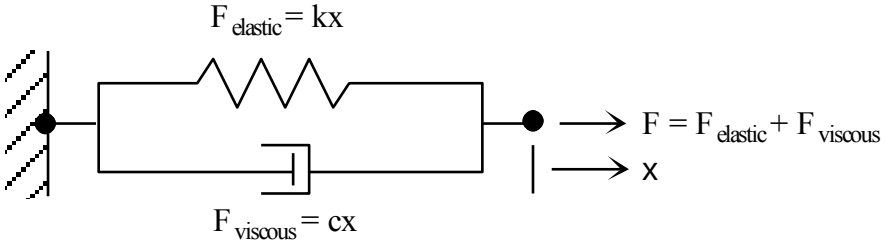
mass = (F) Total mass of the element. No default.

k = (F) Value of spring parameter k. No default.

c = (F) Value of viscous parameter c. No default.

vmax= (F) Conservative estimate of the peak relative, longitudinal velocity of the two end nodes that will occur during the simulation (used to compute stable time step for element). No default.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Viscous element of *linetype* = KELV

PROP (properties - CABL) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=CABL, a linear elastic/plastic spring with compression gap capability.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** CABL subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass k fmax fcutoff gapinitial delt*

linename = (C) Name of the line element model. No default.

mass = (F) Total mass of the element. No default.

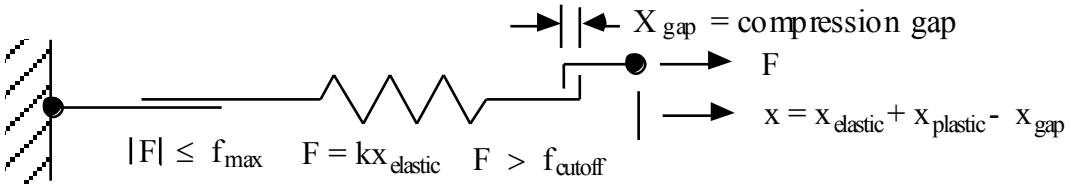
k = (F) Value of spring constant k. No default.

fmax = (F) Peak compressive or tensile force the element may have. Default: *fmax* = 0.0, no plastic limit, element is linear elastic with gap capability.

fcutoff= (F) Compressive force cutoff at which the gap begins to open. Default = 0.0

gapinitial= (F) Initial gap present in the element. Default = 0.0, no gap.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Spring element of *linetype* = CABL

PROP (properties - EOS) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=EOS, a nonlinear equation of state model element.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** EOS subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

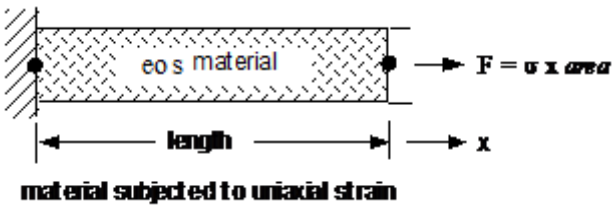
PROP *linename eosname area delt*

linename = (C) Name of the line element model.
No default.

eosname = (C) Name of the continuum material property defining the eos model parameters to use. Must have been previously defined in the **MATR** command group.
No default.

area = (F) Area of the line element. The elements axial force is stress*area.
No default.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model.
Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Uniaxial strain EOS element of *linetype* = EOS

Note: Mass of line element is the mass density of the eos material times the volume of the element, V=length*area.

PROP (properties - CAP) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=CAP, a nonlinear, uniaxial strain, cap model element.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** CAP subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

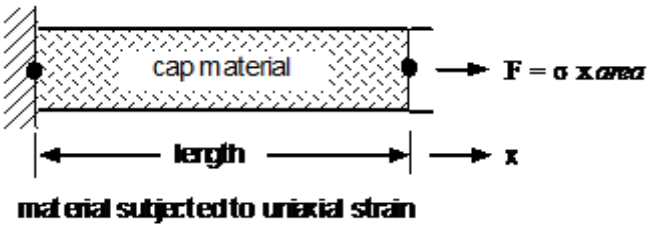
PROP *linename capname area delt*

linename = (C) Name of the line element model.
No default.

capname = (C) Name of the continuum material property defining the cap model parameters to use. Must have been previously defined in the **MATR** command group.
No default.

area = (F) Area of the line element. The elements axial force is stress**area*.
No default.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model.
Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Uniaxial strain soil element of *linetype* = CAP

Note: Mass of line element is the mass density of the cap material time the volume of the element, V=length**area.

PROP (properties - GSPR) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=GSPR, a general spring for any degree-of-freedom whose force displacement behavior is defined by a digitized curve defined by a **CURV** subcommand. The **PROP** command allows the user to configure up to 6 general springs at one time.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** GSPR subcommand and any **CURV** subcommands referenced and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename tmass (rmass) (breakoption)* &
dof1 (scale1) curvname1 ... dofn (scalen) curvnamen delt

linename = (C) Name of the line element model. No default.

tmass = (F) Total translational mass of the element. No default.

rmass = (F) Optional input. The total rotational mass of the element. No default.

breakoption = (C) Optional input. Set to ALL if all springs are to break when any single spring breaks. Otherwise, ignore this input and any spring will break independently of the others. No default.

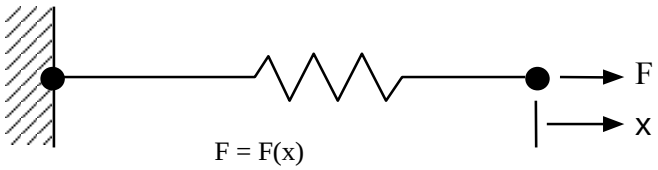
Input up to 6 sets of the following parameters

dof = (C) The local degree of freedom for this spring element. May be any of: X, Y, Z, RX, RY, RZ. No default.

scale= (F) Optional parameter. If input, this parameter scales the force values of the force-displacement data defined by *curvname*. No default.

curvname= (C) Name of the force-displacement data to use to define the behavior of this spring.

delt = (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.



Spring element of *linetype* = GSPR

PROP (properties - EACT) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=EACT, an electostrictive actuator.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** EACT subcommand and precede any **ELEM** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename length area density E delt kappa ps Q prestress bias_volt volt_hist*

linename = (C) Name of the line element model. No default.

length= (F) Length of the element. No default.

area= (F) Cross-sectional area. No default.

density= (F) Mass density. Default = 0.

E= (F) Young's modulus.

delt= (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model. Default: *delt*=0.0, the stable time step is computed based on the model parameters.

kappa= (F) Kappa from Hom's model.

ps= (F) Polarization saturation from Hom's model.

Q (F) Electostrictive coupling constant.

prestress= (F) Prestress (compression negative). Default = 0.0.

bias_volt (F) Bias voltage. Default = 0.

volt_hist= (C) Name of the applied voltage time history (or FUNC) across the element, or OPEN for open circuit conditions. Default = OPEN.

Note: This is a special purpose element that is electrically independent of other actuators, piezoelectric materials, or electric windows declared elsewhere in the model.

PROP (properties - JOINT) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=JOINT, a frictional joint model. Currently restricted to 3D models.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** JOINT subcommand and precede any **SNGL** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename mass delt curvename frcstif cohesion frcangle frcmax & shrsft dilangle resdelta sigtrn dilexp cdamp tdamp*

linename = (C) Name of the line element model. No default.

mass = (F) Total mass of the element, or enter a negative value for mass per unit area Default=0.

delt= (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model.

curvename= (C) Name of the stress-displacement data to use to define the axial behavior of this spring. Note: compression is negative. No default. See note 1.

frcstif= (F) Frictional stiffness. Default=0.

cohesion= (F) Cohesion. Default = 0.

frcangle (F) Tangent of friction angle. Default=0.

frcmax= (F) Maximum value of friction. Default = infinite.

shrsft= (F) Shear softening exponent. Default = 0, no softening. See Note 2 below.

dilangle= (F) Tangent of dilation angle. Default = 0.

resdelta= (F) Shear displacement at which dilation stops. Default = 0.

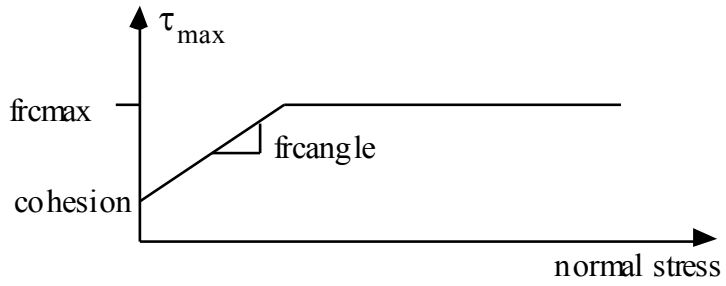
sigtrn = (F) Normal stress at which dilation stops. Default = infinity. Recommended value approximately unconfined compression strength.

dilexp = (F) Dilation exponent. Default = 0; recommended value approximately 4.

cdamp = (F) Fraction of critical dampng in compression. Default = 0

tdamp = (F) Fraction of critical damping in tension. Default = 0.

frcres = (F) Tangent of residual friction angle. Default = *frcang*.



Envelope of allowable shear friction forces between two slideline surface

Notes:

1. Only ELAS or HYST type curves are implemented. The first and last segment slopes define the stress vs. displacement values for displacements beyond the defined range.
2. Shear softening is included by reducing the cohesion & friction angle as a function of irreversible sliding displacement: $c = cohesione^{-shrsft\,\delta}$.
3. Joint dilation is zero when the normal stress exceeds *sigtrn*, or sliding displacement exceeds *resdelta*. Otherwise, the increment of dilation is equal to the increment of sliding displacement multiplied by $dilangle^{*}(1 - sigma / sigtrns)^{dilexp}$. This empirical expression due to Goodman & St. John reduces dilation with increasing normal stress and sliding displacement.
4. For 2D axisymmetric models, element mass should be entered as the mass of an element a radial distance of 1 element length away from the axis of symmetry. The mass of other elements is scaled by the radius.

PROP (properties - BOLT) Subcommand
LINE (line element) Input Group **LINE-PROP**

Purpose: To define properties for a line element model with *linetype*=BOLT.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow a **TYPE** BOLT subcommand and precede any **SNGL or ITFC** subcommand that references the defined line element model.

PCOM SCOM input parameters

PROP *linename material delt area length shearlength*

linename = (C) Name of the line element model. No default.

material= (C) Name of a previously defined ISOH material. No Default.

delt= (F) Time step override for this model. If input, this time step is used as the stable time step for all elements assigned this model.

area= (F) Cross-sectional area. No default.

length= (F) Length of the element. No default.

shearlength= (F) Length of the shear zone. Default = *length*

CURV (curve) Subcommand
LINE (line element) Input Group **LINE-CURV**

Purpose: Specifies a digitized curve which defined the force displacement behavior of a general spring line element (line*type* = GSPR), or the joint element (line*type*=JOINT).

Use Is: Required only if GSPR, or JOINT springs are defined.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must precede any **PROP** subcommands that reference the defined curve.

PCOM SCOM input parameters

CURV *curvname (formoption) (hardoption) disp1 forc1 disp2 forc2 ... dispn forcn*

curvname = (C) Name of this curve. No default.

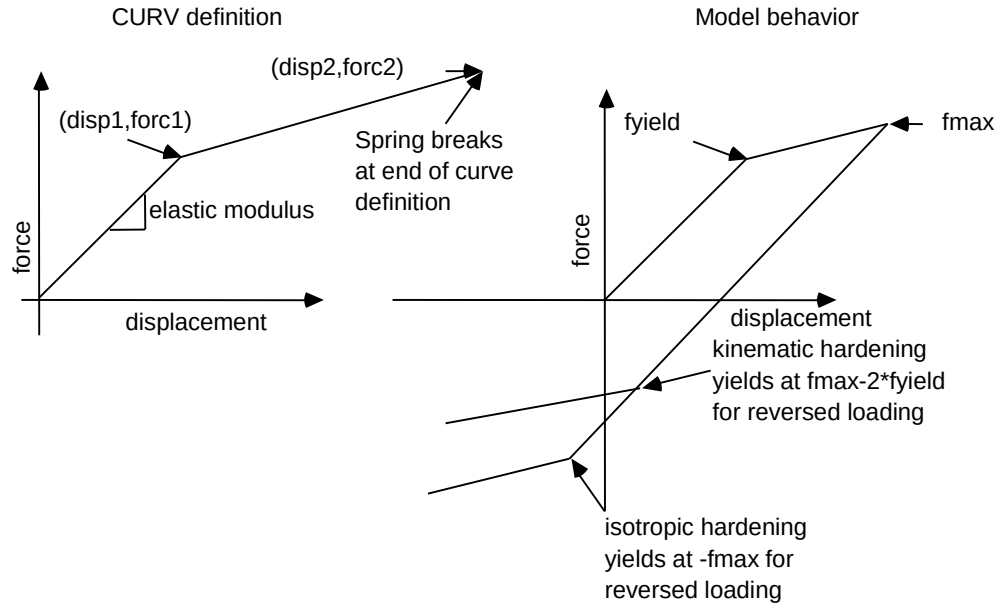
formoption = (C) Optional parameter. If set to STIF, the code interprets the *forc1*, *forc2*, etc values as the stiffness (slope) of the force displacement segment ending at the corresponding displacement value. If not input, the *forc1*, *forc2*, etc values are assumed to be the actual force values corresponding to the defined displacement values. No default. For the JOINT element, stiffness is not implemented, and *forc* is interpreted as stress.

hardoption = (C) Optional parameter. Parameter to define the type of elastic/inelastic behavior which this curve defines. If not input, isotropic hardening is assumed. If input, must be one of: ISOH, KINE, WNDO, ELAS

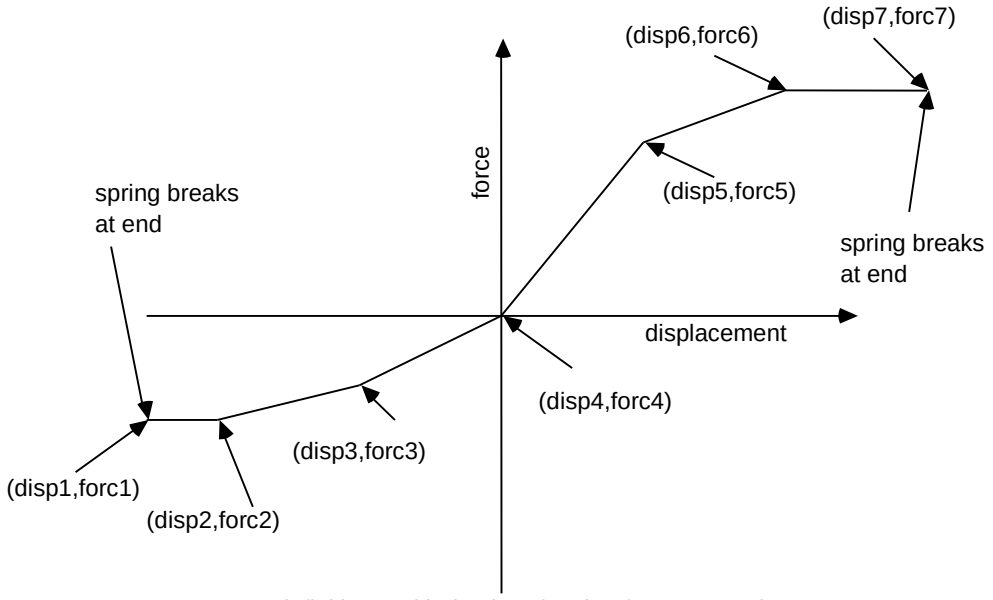
- ISOH = Isotropic hardening plasticity behavior
- KINE = Kinematic hardening plasticity behavior
- WNDO = Force displacement curve representing nonlinear window break.. Note: all springs using a specific WNDO curve break or tear out of the frame at the time the first of these springs reaches the break or tear out displacment.
- ELAS = Elastic behavior exactly follows the input digitized data.
- HYST = Hysteretic behavior. Applies only to JOINT type springs.

The digitized curve must begin with 0.0 or the first postive displacement value for ISOH, KINE or WNDO options. The ELAS option allows the user to define a curve that defines both the negative and positive displacement behavior independently of one another.

dispn, forcn = (F) Displacement-force pairs that define the digitized curve representing the behavior of the spring element. If this curve is assigned to a spring attached to arotational degree-of-freedom, then the displacement values are in radians.



Isotropic and Kinematic hardening options



CURV definition and behavior of spring for ELAS option

LDEF (large displacement) Subcommand
LINE (line element) Input Group **LINE-LDEF**

Purpose: Activates or deactivates large deformation option for general spring line element (line*type* = GSPR) created with following **ELEM** or **SNGL** subcommands. Applies only to models that include the **LDEF** primary command. If not input, no large deformation effects are considered.

Use Is: Effective only for GSPR type line elements.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must precede any **ELEM** or **SNGL** subcommands that generate GSPR type elements for which the **LDEF** option is to apply.

PCOM SCOM input parameters

LDEF *option*

option = (C) Either ON or OFF. No default.

ELEM (elements) Subcommand
LINE (line element) Input Group **LINE-ELEM**

Purpose: To define the connectivity for a string of line elements whose nodes are adjacent to one another and specify the associated line element model.

Use Is: Optional, but either the **ELEM** or **SNGL** subcommand must be input.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow the **PROP** subcommand that defines the line element properties for the defined elements.

PCOM	SCOM	input parameters
	ELEM	*linename (axisname) ibegin iend jbegin jend kbegin kend*

linename = (C) Name of the line element model.
No default.

axisname = (C) Optional input parameter. Used only if *linename* is of type GSPR. The name of the coordinate system which defines the local orientation of the general springs. Set to STND if the local system aligns with the model's global coordinate system.
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the group of line elements defined.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the group of line elements defined.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the group of line elements defined.
Default: *kbegin* = 1, *kend* = maximum K

Note: Only a single string of elements may be defined by a single **ELEM** subcommand. Therefore, if *ibegin* is not equal to *iend*, then *jbegin* = *jend* and *kbegin* = *kend*. Similar constraints apply for the other two cases, i.e., when *jbegin* is not equal to *jend* or when *kbegin* is not equal to *kend*.

ORFC (orifice) Subcommand
LINE (line element) Input Group **LINE-ORFC**

Purpose: To define an orifice connecting 2 EOS line elements.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within LINE Group: None.

PCOM	SCOM	input parameters
	ORFC	*n1 n2 area temperature gas_cons gamma*

n1, n2 = (I) Line elements (in order of definition) connected by the orifice.
No default.

area= (F) Cross-sectional area of the orifice.
No default.

The following 3 parameters limit the flow velocity to Mach 1 for an ideal gas.
Leave blank for other materials.

temperature= (F) Absolute temperature.
No default.

gas-cons= (F) Gas constant.
No default.

gamma= (F) Exponent in the ideal gas law.
No default.

SNGL (single element) Subcommand
LINE (line element) Input Group **LINE-SNGL**

Purpose: To define the connectivity for a single line element and specify the associated line model properties. The element's nodes do not have to be adjacent.

Use Is: Optional, but either the **SNGL** or **ELEM** subcommand must be input.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow the **PROP** subcommand that defines the line element properties for the defined element.

PCOM	SCOM	input parameters
	SNGL	*linename (axisname) i1 j1 k1 i2 j2 k2 (i3 j3 k3 i4 j4 k4 area)*

linename = (C) Name of the line element model.
No default.

axisname = (C) Optional input parameter. Used only if *linename* is of type GSPR. The name of the coordinate system that defines the local orientation of the general springs. Set to STND if the local system aligns with the model's global coordinate system.
No default.

i1, j1, k1 = (I) I, J, and K indices of node 1 of the line element.
No default. For 2D models the K indices should be input as 1.

i2, j2, k2 = (I) I, j, and k indices of node 2 of the line element.
No default. For 2D models the K indices should be input as 1.

i3, j3,k3 = (I) I, j, and k indices of node 3 of the line element.
Used only for linetypes = JOINT or BOLT to define a reference node. No default. See note 1.

I4, j4,k4 = (I) I, j, and k indices of node 4 of the line element.
Used only for linetypes = JOINT or BOLT to define a reference node. No default

area = (F) Cross-sectional area. Used only for linetype = JOINT. Default = 1.0

Notes: 1. The joint element extends from node 1 to node 2. Relative motion between nodes 1 and 2 is separated into normal and tangential components. The normal direction at node 1 is taken as **v3 x v4**. where v3 is the vector from node 1 to node 3, and **v4** is the vector from node 1 to node 4.

2. For 2D models, enter the I, j, and k indices for node 3 *or* node 4 and default the indices for the other. The defaulted node is set to lie in the +K direction from node 1 and the normal and tangential components of motion are determined as in Note 1.

GCON (Single element) Subcommand
LINE (line element) Input Group **LINE-GCON**

Purpose: To define the connectivity for a single line element in general connectivity format and specify the associated line model properties. The element's nodes do not have to be adjacent.

Use Is: Required for General Connectivity models when using **LINE** elements.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow the **PROP** subcommand that defines the line element properties for the defined element.

PCOM SCOM input parameters

GCON *linename (axisname) n1 n2 (n3 n4 area)*

linename = (C) Name of the line element model. No default.

axisname = (C) Optional input parameter. Used only if *linename* is of type GSPR. The name of the coordinate system that defines the local orientation of the general springs. Set to STND if the local system aligns with the model's global coordinate system. No default.

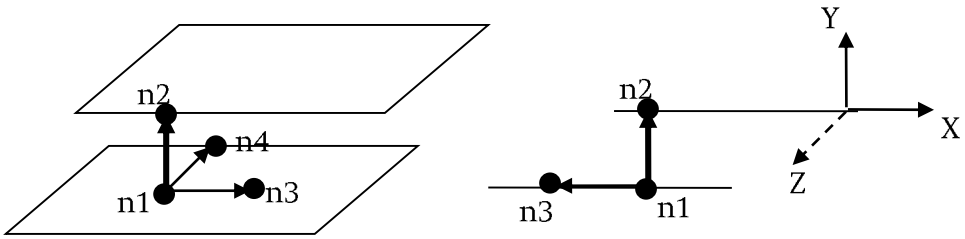
n1 = (I) General connectivity node number of node 1 of the line element. No default.

n2 = (I) General connectivity node number of node 2 of the line element. No default.

n3 = (I) General connectivity node number of node 3 of the line element. Used only for linetypes = JOINT or BOLT to define a reference node. No default. See note 1.

n4 = (I) General connectivity node number of node 4 of the line element. Used only for linetypes = JOINT or BOLT to define a reference node. No default

area = (F) Cross-sectional area. Used only for type = JOINT. Default = 1.



3D Model 2D Model

Notes: 1. The joint element extends from node 1 to node 2. Relative motion between nodes 1 and 2 is separated into normal and tangential components. The normal direction at node 1 is taken as **v3 x v4**. where v3 is the vector from node 1 to node 3, and **v4** is the vector from node 1 to node 4.

2. For 2D models, enter either node 3 or node 4 and default the other. The defaulted node is set to lie in the +Z direction from node 1, and the normal and tangential components of motion are determined as in Note 1.

ITFC (interface) Subcommand
LINE (line element) Input Group **LINE-ITFC**

Purpose: To define the connectivity for a patch of single line elements and specify the associated line model properties. Implemented only for the JOINT and BOLT type elements.

Use Is: Optional, but either the **SNGL** or **ITFC** subcommands must be input.

Multiple Use: Allowed.

Order Dependence within LINE Group: Must follow the **PROP** subcommand that defines the line element properties for the defined element.

PCOM SCOM input parameters

ITFC *linename ib1 ie1 jb1 je1 kb1 ke1 ib2 ie2 jb2 je2 kb2 ke2 xref yref zref & refopt voidopt*

linename = (C) Name of the line element model. No default.

Ib1...ke1 = (I) I, J and K indices of nodes that bound side 1 of the interface. See note 1.

Ib2...ke2 = (I) I, j, and k indices of nodes that bound side 2 of the interface. See note 1.

xref, yref, zref = (F) Coordinates of a reference point relative to side 1. No default.

refoption = (C) Either IN or OUT. If *refopt* = OUT, a vector from side 1 to the reference point points towards side 2. Default = OUT.

voidopt = (C) Either SKIP or OFF. When *voidopt* = SKIP, void elements are skipped during the tributary area calculation for type = JOINT. Default = OFF.

Notes:

1. Each side should be a 2D surface, so *ib1=ie1*, *jb1=je1*, or *kb1=ke1*. Likewise for side 2.
2. This option creates a JOINT type line element from each node on side 1 to the closest node on side 2. It computes tributary areas (for type JOINT) based on the side 1 element faces and assigns reference nodes.
3. For 2D axisymmetric models, the tributary area (for type JOINT) is scaled by the radial distance to the joint. The area is computed as area = $0.5^{*}r_1^{*}dr^{*}d\theta + 0.5^{*}r_2^{*}dr^{*}d\theta$, where $d\theta = 1$ radian, r_1 and r_2 are the radial distances to nodes on either side of the joint, and $dr = |r_2 - r_1|$. There is no scaling of BOLT cross-sectional area.

INFO (interface) Subcommand
LINE (line element) Input Group **LINE-INFO**

Purpose: To print helpful debuggin information to the flxprt file

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within LINE Group: Must follow the subcommands that define the line elements.

PCOM	SCOM	input parameters
	INFO	*option1 option2*

option1 = (C) Enter LENGTH to find the longest line element
No default.

option2 = (C) Enter BOLT to search , BOLT elements, JOINT to search only JOINT elements, or leave blank to search all elements.

PRNT (print) Subcommand
LINE (line element) Input Group **LINE-PRNT**

Purpose: To echo line element information to the flxprt file.

Use Is: Optional

Multiple Use: Allowed.

Order Dependence
within LINE Group: Must follow the subcommands that defining the line elements

PCOM	SCOM	input parameters
	PRNT	

MAGN (magnetostrictive) Input Command

Purpose: To allow the option of including magnetostrictive materials within the model. Currently implemented only for 2D axisymmetric models.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the **GRID** command and precede the **PRCS** command when using the **WNDO**, **NODE**, and **CALC** subcommands.

PCOM	SCOM	input parameters
MAGN		
	WNDO	*ibegin iend jbegin jend kbegin kend*
	DEFN	*coilname*
	NODE	*turns ibegin iend jbegin jend kbegin kend*
	PMAG	*mx my ibegin iend jbegin jend kbegin kend*
	BC	*coilname option histname scalehist shifthist*
	CALC	*option*
	HRGL	*hourglass*
	SIDE	*iside type option value(1) value(2) ... value(6)*
	CONN	*coilname circuitname sourceoption histname scalehist shifthist*
	SLVR	*option*
	END	

WNDO (window) Subcommand
MAGN (Magnetostrictive) Input Group **MAGN-WNDO**

Purpose: To specify the regions of the grid within which the magnetostatic computation is performed. At a minimum, a window must encompass any applied voltage conditions and all magnetostrictive material for which magneto-mechanical coupling should be modeled.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **MAGN** Group: Must precede any **NODE** subcommands.

PCOM	SCOM	input parameters
	WNDO	*ibegin iend jbegin jend kbegin kend*

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the magnetic window.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the magnetic window.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the magnetic window.
Default: *kbegin* = 1, *kend* = maximum K

Notes:
1. The larger the magnetic window, the longer it takes to compute the magnetostatic solution at each timestep.
2. More than one magnetic window can be declared; the windows, however, must not overlap or contact each other and they will be magnetically decoupled from one another.
3. Several data arrays will be created for all nodes or elements within the magnetic window. The i - index of these arrays ranges from 1 to the total number of nodes (or elements) summed over all windows. These are:
   BX(i) - the x-component of each element's magnetic field
   BY(i) - the y-component of each element's magnetic field
   BZ(i) - the z-component of each element's magnetic field

   Note that the BX, BY, and BZ arrays are available only if the user request them with the **CALC** subcommand. The BZ array is available only for 3D models.

DEFN (electrode definition) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-DEFN**

Purpose: To initiate the definition of coils within the model. This definition is used when computing the magnetostatic field. The definition is completed by using **NODE** subcommands to define the grid nodes that make up the coil and the **CONN** subcommand to connect it to a circuit model (if any) or the **BC** subcommand to define current boundary conditions directly to the coil.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAGN** Group: Must follow the **WNDO** subcommand and precede any subcommands that specify the grid nodes that make up this coil and any **CONN** or **BC** subcommands that reference this coil.

PCOM	SCOM	input parameters
	DEFN	*coilname*

coilname = (C) The name of this coil. No default.

Notes: 1. An example of a group of commands that define a coil and prescribea driving source for a 2D model:

```
MAGN
    WNDO  1  20  1 20        /* define the magnetic window
C  begin definition of coil
    DEFN C1
    NODE 1  20  5 6  /* specify coil  nodes
C  connect bottom electrode to circuit C1 defined by previous CIRC commands
C  and define the driving function.
    BC  C1 VOLT FUNC
    END
```

2. Information about each defined coil is stored within the data array MAGE(i,j). This array is doubly subscripted. The j-index ranges from 1 to the number of coils specified for the model. The i-index ranges from 1 to 5 where:

i=1 - voltage across the coil at the present time
i=2 - total charge (integral of i=3) the present time
i=3 - current , time rate of change of charge over the last time step
i=4 - temporary work slot
i=5 - temporary work slot

Example: Request time histories of potential voltage and charge for the third and fourth coils defined for the model.

```
POUT
    RATE 1
    HIST MAGE  1  2  1  3  4  1        /*  four time histories will be saved
    END
```

NODE (coil nodes) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-NODE**

Purpose: To select nodes of the model that belong to the coil defined by the last **DEFN** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAGN** Group: Must follow the **DEFN** subcommand that initiated the definition of the coil.

PCOM	SCOM	input parameters
	NODE	*turns ibegin iend jbegin jend kbegin kend*

turns = (F) Number of turns in the area defined by this node command.
Default: 0.

ibegin, iend = (I) Beginning and ending I-indices of nodes that will be assigned to this coil.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that will be assigned to this coil.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that will be assigned to this coil.
Default: *kbegin* = 1, *kend* = maximum K

Note: More than one **NODE** subcommand may be used to identify a coil's nodes.

PMAG (permanent magnet) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-PMAG**

Purpose: To define a permanent magnet.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MAGN** Group: None.

PCOM	SCOM	input parameters
	PMAG	*mx my ibegin iend jbegin jend kbegin kend*

mx,my = (F) Magnetization in the x and y directions respectively. Default: 0.

ibegin, iend = (I) Beginning and ending I-indices of nodes that will be assigned to this magnet.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that will be assigned to this magnet.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that will be assigned to this magnet.
Default: *kbegin* = 1, *kend* = maximum K

Note: Permanent magnets are “turned on” at timestep 1.

BC (coil boundary conditions) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-BC**

Purpose: To specify current boundary conditions to be applied directly to a coil. If there is a circuit connected to a coil, the driving source is defined by the **CONN** (connect) subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAGN** Group: Must follow the **DEFN** subcommand that defines the coil.

PCOM	SCOM	input parameters
	BC	*coilname option histname scalehist shifthist*

coilname = (C) Name of the coil to which this boundary prescription applies. No default.

option = (C) Boundary condition option.
Any of: CRNT= prescribed current
VOLT= prescribed voltage

histname = (C) Name of the time history that defines the prescribed constraint. Set to FUNC if the **FUNC** command is used to define the time history. Set to the name of time history if the **DATA HIST** command is used to define a digitized time history. No default.

scalehist = (F) Scale factor to scale the prescribed voltage or charge defined by *histname* before applying it to the electrode.
Default = 1.0

shifthist = (F) Shift value to add to the time of *histname* before applying the time history to the electrode.
Default = 0.0

Notes: 1. If *scalehist* is -1.0, the sign of the prescribed voltage is reversed. If *scalehist* is 3.0, the magnitude of the prescribed voltage is three times greater than in *histname*.
2. If *histname* defines a square pulse that begins at time 0.0, setting *shifthist* to 1.0e-6 would cause the prescribed square pulse signal to begin at time = 1.0e-6.

CALC (calculate) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-CALC**

Purpose: To request that the B field be computed within the magnetic window(s).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAGN** Group: None.

PCOM SCOM input parameters

CALC *option*

option = (C) Set to B to request that the B field be computed throughout the computation.
No default.

Note: The B field for each element within the magnetic window will be computed and stored in the BX(i,j,k), BY(i,j,k), and BZ(i,j,k) data arrays, where the i-, j- and k-indices are the local elemental indices within the electric window. The BZ array is available only for 3D models.

HRGL (hourglass) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-HRGL**

Purpose: To set the hourglass control value.

Use Is: Optional. If not input the hourglass supression constant is set to 1.0.

Multiple Use: Allowed.

Order Dependence
within **MAGN** Group: None.

PCOM	SCOM	input parameters
	HRGL	*hourglass*

hourglass = (F) Hourglass control constant. Value should be kept less than or equal to 1.0. Default = 1.0

SIDE Subcommand
MAGN (magnetostrictive) Input Group **MAGN-SIDE**

Purpose: To assign boundary conditions to an entire side of an magnetic window.

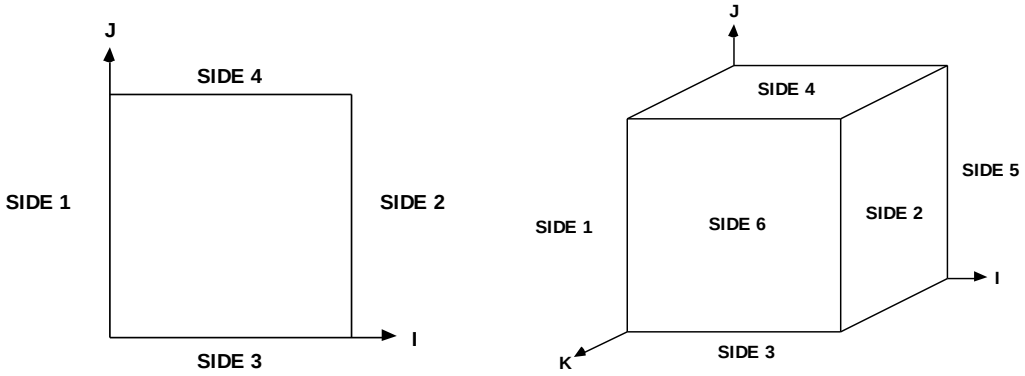
Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAGN** Group: Must follow the **WNDO** subcommand defining the magnetic window to which the boundary conditions apply. Must precede other **WNDO** subcommands.

PCOM	SCOM	input parameters
	SIDE	*iside type option value(1) value(2) ... value(6)*

iside = (I) The side number specifying which side to apply the specified boundary condition on. Must be from 1 to 6 as shown.



2Dimensional models 3Dimensional models

Side number relationship to grid indices orientation

type = (C) Type of boundary conditions to be applied on side *iside*. No default.
Any of: INFN = infinite element

option= (C) Option for the type of boundary condition specified. Valid choices for *option* depend on the choice of *type*.

Boundary type	options
INFN	RADL — element diverges radially NRML — element projects normally

value(n)= (F) For INFN boundaries, *value(1), value(2), value(3)* are the (x,y,z) coordinates of the reference point (Pole). Default = center of electric window.

Notes: 1. The default boundary conditions on each magnetic window are that the normal component of the electric field is zero.
2. The INFN boundary condition uses "infinte elements" to terminate the grid. These approximate an infinite domain by enforcing a decay of the potential to zero at infinity. For option = RADL, the decay is radial from the reference point (e.g., for a small or point source). For option = NRML, the decay is normal to the boundary (e.g., for a large source).
3. The INFN conditions should not be used on the axis of an axisymmetric model.

CONN(connect) Subcommand
MAGN (magnetostrictive) Input Group **MAGN-CONN**

Purpose: To connect a circuit and a source to a coil.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **CIRC** command group that defines the circuit and the **DEFN** subcommand that defines the coil being connected to the circuit.

PCOM SCOM input parameters

CONN *coilname circuitname sourcetype histname scalehist shifthist*

coilname = (C) Coil name. No default.

circuitname = (C) Circuit name. No default.

sourcetype = (C) Source type. No default.
Any of: VOLT = prescribed voltage
CHRG = prescribed charge
GRND = ground
OPEN = open circuit

histname = (C) Name of time history that defines applied voltage or charge at the source. Not input for GRND or OPEN options. Set to FUNC if the **FUNC** command is used to define the time history. Set to the time history name if the **DATA HIST** command is used to define a digitized time history. No default.

scalehist = (F) Scale factor to scale the prescribed voltage or charge defined by *histname* before applying it to the circuit. Not input for GRND or OPEN options. Default = 1.0.

shifthist = (F) Shift value to add to the time of *histname* before applying it to the circuit. Not input for GRND or OPEN options. Default = 0.0.

Notes: 1. Source types may be changed during the course of a calculation (e.g., a prescribed voltage may be switched to an open circuit). This discontinuous action can introduce unwanted noise into the calculation if not carefully timed.

2. Circuit element definitions may also be changed during the course of a calculation (e.g., for send or receive modes). This is done by defining a second circuit with the same number of elements as the original circuit, then issuing a **MAGN CONN** command to connect the coil to the second circuit at the desired time. Again, this discontinuous action can introduce unwanted noise unless the circuit is nearly quiet.

3. The charge and voltage throughout the circuits are available in the CRTQ and CRTV data arrays, respectively. These arrays are dimensioned 1 to the number of circuit nodes. There are two nodes for each circuit element. These are ordered so the node closest to the transducer appears first. A table in the flxprt file following the **PIEZ** command group lists the circuit elements and the corresponding CRTQ, CRTV indices. Time histories of these arrays may be requested with the **POUT** command group. Note that the voltages on connected nodes are identical, but charges on connected nodes sum to zero.

SLVR (solver) Subcommand
MAGN Input Group **MAGN-SLVR**

Purpose: To define the time integration scheme for magnetostatic solutions.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM SCOM input parameters

SLVR *slvrtype alpha droptol*

slvrtype (C) Solver type. Any of: DRCT, DISK, CGDS, CGIC or SPLU. Default = DRCT.

alpha (F) Time integration parameter for generalized trapezoid rule. Default = 0.5

droptol (F) Drop tolerance for SPLU solver (see note 3 below). Default = 1.e-12

Notes: 1. The DRCT option implements an in-core solver. This is the preferred solver if the problem is small enough to fit in available RAM. The DISK option implements an out-of-core direct solver. This is somewhat slower than the in-core solver, but is limited only by available disk, not RAM. CGDS is an iterative method using conjugate gradients with diagonal scaling. It is the most efficient method in terms of RAM requirements, and does not require matrix factorization. CGIC implements conjugate gradients with an incomplete cholesky preconditioner. CGIC should be considered experimental for now because the preconditioner is recomputed each timestep. SPLU is a sparse direct solver

2. For > 0.5, the time integration is unconditionally stable. Optimal accuracy is achieved at = 0.5, but stability characteristics improve as is increased towards 1.0.
3. The SPLU solver tries to preserve sparsity during LU factorization. First the matrix is normalized to 1.0. During factorization, entries which become smaller than the drop tolerance are removed. Larger values reduce storage requirements, but reduce accuracy.

MASS Input Command

MASS

Purpose: To allow the user to add mass to a grid node or redefine the mass of a node.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence:Must follow the **GRID** command and precede the **PRCS** command.

PCOM	SCOM	input parameters
MASS		
	ADD	*masstype value i-index j-index k-index*
	SET	*masstype value i-index j-index k-index*
	GCON	*option masstype nodegc value*
	PRNT	
	END	

Note: The mass of a node may be modified by more than one **ADD** or **SET** subcommand. Overlapping modifications are performed in the order input and are accumulative.

ADD Subcommand
MASS Input Group

MASS-ADD

Purpose: To add a contribution to the mass of a grid node.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MASS** Group: None.

PCOM	SCOM	input parameters
	ADD	*masstype value i-index j-index k-index*

masstype = (C) Type of mass to add a contribution to.
Any of: TRAN= translational mass (all components)
ROT= rotational mass (all components)
No default.

value = (F) Value of the mass to add.
Default = 0.0

i-index, j-index,k-index = (I) The i-, j- and k-indices of the node whose mass is being modified.
No default.

SET Subcommand
MASS Input Group

MASS-SET

Purpose: To set the mass of a grid node to a specified value.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MASS** Group: None.

PCOM	SCOM	input parameters
	SET	*masstype value i-index j-index k-index*

masstype = (C) Type of mass being modified.
Any of: TRAN= translational mass (all components)
ROT= rotational mass (all components)
No default.

value = (F) Value of the nodes mass.
Default = 0.0

i-index, j-index,k-index = (F) The i-, j- and k-indices of the node whose mass is being modified.
No default.

GCON (general connectivity) Subcommand
MASS Input Group **MASS-GCON**

Purpose: To add to or set the value of the mass of a grid node using general connectity syntax for node mapping and mass scaling.
See Section 2: *Importing General Connectivity Models*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MASS** Group: None.

PCOM SCOM input parameters

GCON *option masstype nodegc value*

option = (C) Type of action to perform.
Any of: ADD= add mass contribution to the node
SET= set the value of the mass for the node
No default.

masstype = (C) Type of mass being modified.
Any of: TRAN= translational mass (all components)
ROT= rotational mass (all components)
No default.

value = (F) Value of the mass being added or set.
Default = 0.0

nodegc = (I) Local general connectivity node number whose mass will be modified with this command.
No defaults.

PRNT Subcommand
MASS Input Group **MASS-PRNT**

Purpose: To print a list of all nodes modified by the **MASS** command group..

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MASS** Group: None.

PCOM	SCOM	input parameters
	PRNT	

MATR (material) Input Command

Purpose: To define material properties for each continuum material in the model.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence: Must precede the **SITE** command.

PCOM SCOM input parameters

MATR

TYPE	*mattype*
PROP	*matname density bulk shear hourglas damplinear dampquad* [all other constitutive model parameters follow] [see specific model types for parameters]
HRGL	*matname keyword(s)*
IMPD	*matname cp cs*
DAMP	*matname volrate*
WVSP	*option*
VDMP	*matname centfreq option dilatation shear frequency expv distance exps*
SDMP	*matname centfreq option dilatation shear frequency expv distance exps*
MDMP	*matname centfreq option damping wavetype frequency expv distance exps*
RDMP	*matname centfreq option dilatation shear frequency expv distance exps*
SLOSS	*matname mechfactr dielfactr piezfactr*
FRNG	*frqmin frqmax wtshear*
PRNT	*option matname frqmin frqmax frqinc (filename)*
NWTN	*matname volumetric shear*
AXIS	*matname axisname*
AFRC	*matname fractabsorb*
PIEZ	*matname irow1 jcol1 value1 irow2 jcol2 value2 ... irown jcoln valuen*
ELEC	*matname epsilonx epsilony epsilonz*
THEX	*matname epsilonx epsilony epsilonz*
STOR	*option*

PCOM	SCOM	input parameters
PERM		*matname (option) permx permy permz*
MGST		*matname (option) irow1 jcol1 value1 irow2 jcol2 value2 ... irown jcoln valuen*
ELST		*matname psat kappa epsilon0 q111 q211*
RATE		*matname betz expnt edot1 factor1 edot2 factor2 ... edotn factorn*
THRM		*matname C kx' ky' kz' hgls cb wb Tb*
COPY		*copyfrom copyto*
CURV		*dataname nvalues filename scalex scaley shiftx (option)*
PARM		*store_option*

Note: The VOID material type (with null material properties) is automatically appended to the group of user input materials.

TYPE Subcommand
MATR (material) Input Group **MATR-TYPE**

Purpose: To define the type of constitutive model for the following **PROP** subcommands.

Use Is: Optional. If no **TYPE** subcommand is input, a linear elastic material model is assumed for all **PROP** subcommands.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must precede all **PROP** subcommands of the type specified.

PCOM SCOM input parameters

TYPE *mattype*

mattype = (C) Type of constitutive model to assign to any following **PROP** subcommands.

Any of:

ELAS=	linear elastic material
CAP=	2 invariant cap model for soil and rock
VCAP=	2 invariant viscoelastic cap model
ESCP=	effective stress (2-phase) cap model
PLCP=	piecewise linear cap model
CONC=	3 invariant concrete model
CAP3=	3 invariant model for soil, rock and concrete
VPLS=	3 invariant viscoplastic model
BCAV=	linear elastic material with J1 tension cutoff and reversible tension behavior
LETI=	linear elastic, transversely isotropic
LEAN=	linear elastic, anisotropic
ISOH=	isotropic hardening von Mises model, (not included for 1D models)
SFT1=	3 invariant, softening concrete model
SFT2=	3 invariant model with embedded crack for concrete and rock.
EOS=	nonlinear equation of state model
TISU=	B/A nonlinear tissue model
YEOH=	Yeoh's Hyperelastic form
PLIH=	piecewise Linear Isotropic Hardening

No default.

PROP (properties - ELAS) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = ELAS.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** ELAS subcommand or precede any other **TYPE** subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

PROP (properties - CAP) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = CAP.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** CAP subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype a b c d w r
tcut geop xint xntn ttyp anon

matname =	(C)	Material name. No default.
density =	(F)	Mass density. No default.
bulk =	(F)	Bulk modulus. No default.
shear =	(F)	Shear modulus. No default.
hourglas =	(F)	Hourglass suppression coefficient. Default = .01
damplinear =	(F)	Linear coefficient for artificial viscosity. Default = 0.0
dampquad =	(F)	Quadratic coefficient for artificial viscosity. Default = 0.0
rltype =	(F)	Option flag: soil = 1.0, rock = 2.0, elastic = -1.0, no cap = -2.0. No default.
a, b, c =	(F)	Failure surface parameters. No default.
d, w =	(F)	Cap-hardening parameters. No default.
r =	(F)	Cap shape. No default.
tcut =	(F)	Tension cutoff. Default = 0.0
geop =	(F)	Geostatic pressure. Default = 0.0
xint =	(F)	Cap intercept of J1 axis. Default = 0.0
xntn =	(F)	Reversible tension if > 0.0. Default = 0.0
ttyp =	(F)	Tension surface coefficient. Default = 0.0
anon =	(F)	Volumetric associativity. $0 \leq anon \leq 1$. Default = 1.0 (fully associated).

PROP (properties - VCAP) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = VCAP.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** VCAP subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype a b c d w r
tcut geop xint xntn ttyp tauv taud
rksf rgsf

matname =	(C)	Material name. No default.
density =	(F)	Mass density. No default.
bulk =	(F)	Bulk modulus. No default.
shear =	(F)	Shear modulus. No default.
hourglas =	(F)	Hourglass suppression coefficient. Default = .01
damplinear =	(F)	Linear coefficient for artificial viscosity. Default = 0.0
dampquad =	(F)	Quadratic coefficient for artificial viscosity. Default = 0.0
rltype =	(F)	Option flag: soil = 1.0, rock = 2.0, elastic = -1.0, no cap = -2.0. No default.
a, b, c =	(F)	Failure surface parameters. No default.
d, w =	(F)	Cap-hardening parameters. No default.
r =	(F)	Cap shape. No default.
tcut =	(F)	Tension cutoff. Default = 0.0
geop =	(F)	Geostatic pressure. Default = 0.0
xint =	(F)	Cap intercept of J1 axis. Default = 0.0
xntn =	(F)	Reversible tension if > 0.0. Default = 0.0
ttyp =	(F)	Tension surface coefficient. Default = 0.0
tauv =	(F)	Relaxation time for volumetric strains. No default.
taud =	(F)	Relaxation time for deviatoric strains. No default.
rksf =	(F)	Ratio of slow to fast bulk modulus. No default.
rgsf =	(F)	Ratio of slow to fast shear modulus. No default.

PROP (properties - ESCP) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = ESCP, an effective stress (2-phase) cap model. The soil skeleton is represented with the standard 2-invariant cap model and the fluid with a nonlinear pressure volume relation.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** ESCP subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype a b c d w r
tcut geop xint xntn ttyp phi0 blks
n e f tenf umax sat

matname =	(C)	Material name. No default.
density =	(F)	Mass density. No default.
bulk =	(F)	Bulk modulus. No default.
shear =	(F)	Shear modulus. No default.
hourglas =	(F)	Hourglass suppression coefficient. Default = .01
damplinear =	(F)	Linear coefficient for artificial viscosity. Default = 0.0
dampquad =	(F)	Quadratic coefficient for artificial viscosity. Default = 0.0
rltype =	(F)	Option flag: soil = 1.0, rock = 2.0, elastic = -1.0, no cap = -2.0. No default.
a, b, c =	(F)	Failure surface parameters. No default.
d, w =	(F)	Cap-hardening parameters. No default.
r =	(F)	Cap shape. No default.
tcut =	(F)	Tension cutoff. Default = 0.0
geop =	(F)	Geostatic pressure. Default = 0.0
xint =	(F)	Cap intercept of J1 axis. Default = 0.0
xntn =	(F)	Reversible tension if > 0.0. Default = 0.0
ttyp =	(F)	Tension surface coefficient. Default = 0.0
phi0 =	(F)	Initial porosity. No default
blks =	(F)	Bulk modulus of solids. No default.
n, e, f =	(F)	Fluid equation of state parameters. No default.
tenf =	(F)	Bulk modulus of fluid in tension. No default.
umax =	(F)	Upper bound on expected fluid pressure. No default.
sat =	(F)	Saturation. No default.

PROP (properties - PLCP) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = PLCP (piecewise linear cap model).

Use Is: Optional.

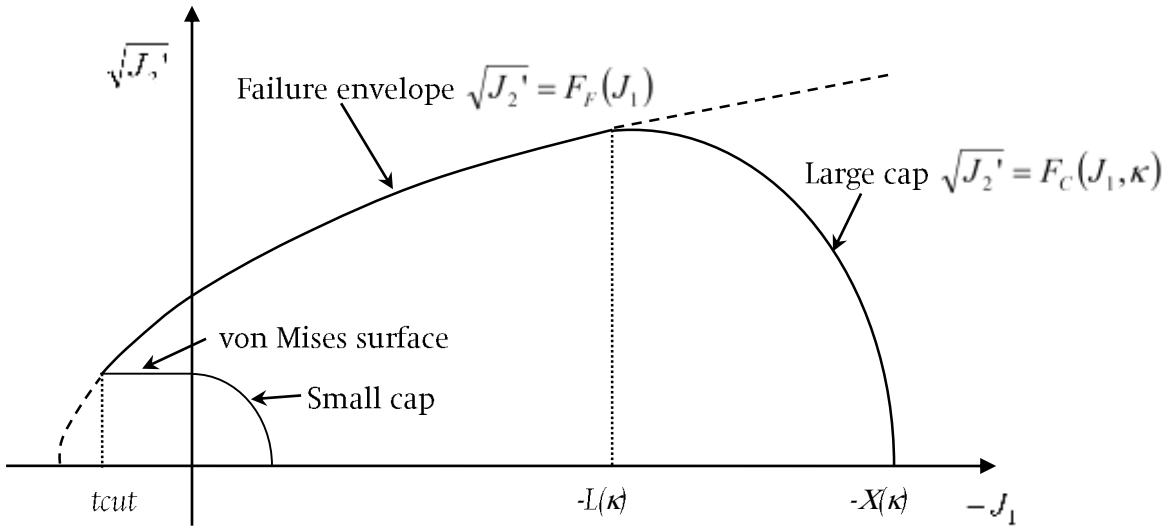
Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** PLCP subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad rltype xint geop anon xntn tcut npairs failsurf caphard capshape & beta bexp*
edot(1) sig(1) edot(2) sig(2) ...edot(npairs) sig(npairs)

matname =	(C)	Material name. No default.
density =	(F)	Mass density. No default.
bulk =	(F)	Bulk modulus. No default.
shear =	(F)	Shear modulus. No default.
hourglas =	(F)	Hourglass suppression coefficient. Default = .01
damplinear =	(F)	Linear coefficient for artificial viscosity. Default = 0.0
dampquad =	(F)	Quadratic coefficient for artificial viscosity. Default = 0.0
rltype =	(F)	Option flag: soil = 1.0, rock = 2.0, elastic = -1.0, no cap = -2.0. No default.
xint =	(F)	Cap intercept of J1 axis. Default = 0.0 See Note 1.
geop =	(F)	Geostatic pressure. Default = 0.0
anon =	(F)	Volumetric associativity. $0 \leq anon \leq 1$. Default = 1.0 (fully associated).
xntn =	(F)	Reversible tension if > 0.0. Default = 0.0
tcut =	(F)	Tension cutoff. Default = 0.0
npairs =	(F)	Number of points ($\leq$ 20) defining rate dependent behavior in uniaxial compression. Enter 0 to disable rate effects.
failsurf =	(C)	Name of previously defined failure surface curve. No default. See Notes 2 and 3.
caphard =	(C)	Name of previously defined cap hardening curve. No default. See Notes 2 and 4.
capshape =	(C)	Name of previously defined cap shape curve. No default. See Notes 2 and 5.
beta =	(F)	Cap rate coefficient. Default = 0.
bexp =	(F)	Cap rate exponent. Default = 0.
edot, sig =	(F)	Pairs of plastic strain rate vs. axial stress in unconfined compression. These pairs all go on one line.



Notes: 1. *xint* = -*X*(κ) is the initial cap intercept of the J_1 axis.

2. Use the DATA HIST command to enter the piecewise linear curves as Y vs.. X, with X in the first column and Y in the second column. The NCLE option must be used to allow decreasing curves.

3. The failure surface curve should be entered as pairs of $\sqrt{J_2'}$ vs.. J_1, where J_1 = -3*P*. J_1 values are entered as negative. $\sqrt{J_2'}$ should be a monotonically decreasing function of J_1 (increasing with pressure).

4. The cap hardening curve should be entered as pairs of ε_v^p vs.. *X*(κ) , where *X*(κ) is the cap intercept of the J_1 axis and ε_v^p is the volumetric plastic strain (negative in compression). *X*(κ) are entered as negative values.

5. The cap shape curve should be entered as pairs of *R* vs.. *L*(κ), where *L*(κ) defines the lower J_1 range of the cap and *R* is the cap shape defined as the ratio of the cap height to width on the J_1, $\sqrt{J_2'}$ axes. *L*(κ) are entered as negative values.

6. For many geologic materials, the slope of the uniaxial compressibility curve appears to increase with increasing strain rate. *Beta, bexp* adjust the baseline rate enhancement on the cap to include this effect.

PROP (properties - CONC) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = CONC.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** CONC subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
f'_c *az au xi r1 r2 d*
w r δ *xint*

matname = (C) Material name. No default.
density = (F) Mass density. No default.
bulk = (F) Bulk modulus. No default.
shear = (F) Shear modulus. No default.
hourglas = (F) Hourglass suppression coefficient. Default = .01
damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0
dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0
f'_c = (F) Unconfined compression strength. No default.
az = (F) Failure surface parameter. Default = .1
au = (F) Failure surface parameter. Default = 1.2
xi = (F) Failure surface parameter. Default = 3.67
r1 = (F) Failure surface parameter. Default = 1.59
r2 = (F) Failure surface parameter. Default = 1.94
d, w = (F) Cap-hardening parameters. No default.
r = (F) Cap shape. No default.
δ = (F) Corner parameter. Default = 0.0
xint = (F) Cap intercept of J1 axis. Default = 0.0

PROP (properties - CAP3) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = CAP3.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** CAP3 subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype p1 p2 p3 p4 p5 p6
pcut b1 xio cr bf2 cw cd
xint elstrt geop cw1 cd1 cap5

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

rltype = (F) Option flag. Either 6.0 or 7.0. No default.

p1 = (F) Parameter f'_c if *rltype* = 6, *c0* if *rltype* = 7. No default.

p2 = (F) Parameter *az* if *rltype* = 6, *c1* if *rltype* = 7.

p3 = (F) Parameter *au* if *rltype* = 6, *c2* if *rltype* = 7.

p4 = (F) Parameter *xi* if *rltype* = 6, *d0* if *rltype* = 7.

p5 = (F) Parameter *r1* if *rltype* = 6, *d1* if *rltype* = 7.

p6 = (F) Parameter *r2* if *rltype* = 6, *d2* if *rltype* = 7.

pcut = (F) Pressure cutoff.

b1 = (F) Parameter not used.

xio = (F) Always use internal default.

cr = (F) Cap shape.

bf2 = (F) Parameter not used.

cw = (F) Cap-hardening parameter.

cd = (F) Cap-hardening parameter.

xint = (F) Initial cap position.

elstrt = (F) Always use internal default.

geop = (F) Geostatic pressure.

cw1 = (F) Parameter not used.

cd1 = (F) Cap-hardening rule.

cap5 = (F) Cap type: 0=default, 1=similar to Failure surface

rltype = 6.0
Default internally computed

rltype = 7.0
No default

PROP (properties - VPLS) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = VPLS.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** VPLS subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype p1 p2 p3 p4 p5 p6
gama b1 xio cr bf2 cw cd
xint elstrt geop cw1 cd1 cap5

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

rltype = (F) Option flag. Either 6.0 or 7.0. No default.

p1 = (F) Parameter f'_c if *rltype* = 6, *c0* if *rltype* = 7. No default.

p2 = (F) Parameter *az* if *rltype* = 6, *c1* if *rltype* = 7.

p3 = (F) Parameter *au* if *rltype* = 6, *c2* if *rltype* = 7.

p4 = (F) Parameter *xi* if *rltype* = 6, *d0* if *rltype* = 7.

p5 = (F) Parameter *r1* if *rltype* = 6, *d1* if *rltype* = 7.

p6 = (F) Parameter *r2* if *rltype* = 6, *d2* if *rltype* = 7.

gama = (F) Viscosity parameter.

b1 = (F) Flow function parameter.

xio = (F) Always use internal default.

cr = (F) Cap shape.

bf2 = (F) Tension surface parameter.

cw = (F) Hardening parameter.

cd = (F) Hardening parameter.

xint = (F) Initial cap position.

elstrt = (F) Always use internal default.

geop = (F) Geostatic pressure.

cw1 = (F) Parameter not used.

cd1 = (F) Cap-hardening parameter.

cap5 = (F) Parameter not used.

rltype = 6.0

Default internally computed

rltype = 7.0

No default

PROP (properties - BCAV) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = BCAV.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** BCAV subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad pcut reduce*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

pcut = (F) Mean pressure cutoff. Default = 0.0

reduce = (F) Reduction factor. Element undergoing tension cutoff will have its stress reduced (*reduce* x *pcut*) each time step. Default = 0.0

PROP (properties - LETI) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = LETI, Linear Elastic Transversely Isotropic

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** LETI subcommand.

PCOM SCOM input parameters

PROP *matname density dumybulk dumyshear hourglas damplinear dampquad E E′ ν ν′ G′ n1 n2 n3*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

dumybulk = (F) Dummy bulk modulus should be default. Any input value is replaced by value computed from stiffness constants below. Used only for hourglass suppression or stiffness proportional damping.

dumyshear = (F) Dummy shear modulus should be default. Any input value will be replaced by value computed from stiffness constants below. Used only for hourglass suppression or stiffness proportional damping.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

E= (F) Young's modulus in isotropic plane.

E′= (F) Young's modulus perpendicular to isotropic plane.

ν= (F) Poisson's ratio in isotropic plane.

ν′= (F) Poisson's ratio perpendicular to isotropic plane.

G′= (F) Shear modulus perpendicular to isotropic plane.

n1, n2, n3= (F) Normal to isotropic plane.

Notes: 1. This model is applicable only to small rotation problems, as stiffness properties are not rotated throughout the computation.
2. The values of E, E', v, v', G' must lead to a positive definite stiffness matrix; no internal checks are performed by the code.
3. An alternate specification of elastic constants is provided by the LEAN material type.

PROP (properties - LEAN) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = LEAN, Linear Elastic Anisotropic

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** LEAN subcommand.

PCOM SCOM input parameters

PROP *matname density dumybulk dumyshear hourglas damplinear dampquad*
(option) c11 c12 c13 c14 c15 c16 c22
c23 c24 c25 c26 c33 c34 c35
c36 c44 c45 c46 c55 c56 c66

matname = (C) Material name. No default.

density = (F) Mass density. No default.

dumybulk = (F) Dummy bulk modulus should be default. Any input value will be replaced by value computed from stiffness constants below. Used only for hourglass suppression or stiffness proportional damping.

dumyshear = (F) Dummy shear modulus should be default. Any input value will be replaced by value computed from stiffness constants below. Used only for hourglass suppression or stiffness proportional damping.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

option= (C) Optional input parameter defining the meaning of c(i,j). Any of:
STIF = stiffness constants
CMPL = compliance constants
If not input, stiffness constants are assumed.

c_{ij}= (F) Elastic stiffness or compliance coefficients depending on *option* above. See Note 5 below.

Notes:
1. The stiffness properties input must lead to a positive definite stiffness matrix. The code issues an error message if the input values do not satisfy this requirement.
2. The CFL stability calculation is approximate in that it calculates wavespeed at a discrete set of samples (every 10 degrees of solid angle) and picks the maximum wavespeed from this set. If stability problems arise, try lowering the stability factor slightly.
3. c(i,j) are defined by the following convention where γ_{ij} denote engineering values of shear strain (i.e., twice the tensor values)

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix} = \begin{bmatrix} c11 & c12 & c13 & c14 & c15 & c16 \\ c12 & c22 & c23 & c24 & c25 & c26 \\ c13 & c23 & c33 & c34 & c35 & c36 \\ c14 & c24 & c34 & c44 & c45 & c46 \\ c15 & c25 & c35 & c45 & c55 & c56 \\ c16 & c26 & c36 & c46 & c56 & c66 \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix}$$

4. If compliances are input, c(i,j) are NOT the actual compliance tensor values but are as defined in Auld below. Their values are: the same as the tensor values for terms having neither subscript greater than 3, twice the tensor values for terms with a single subscript greater than 3, and four times the tensor values when both subscripts are greater then 3. See Auld, B.A., *Acoustic Fields and Waves in Solids*, R.E. Krieger Publishing Co., Malabar, Fl., 1990., p66.

PROP (properties - ISOH) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = ISOH.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** ISOH subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad sigyield Eyield sigmax npairs (Keyword params) edot(1) sig(1) edot(2) sig(2) ...edot(npairs) sig(npairs)*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

sigyield = (F) Yield stress in uniaxial stress test. No default.

Eyield = (F) Material modulus after yield.
Default = 0.0, perfectly plastic behavior after yielding.

sigmax = (F) Maximum stress in uniaxial stress test.
Default = *sigyield*.

npairs= (F) Number of points ($\leq$ 10) defining rate dependent behavior in uniaxial tension. Set to 0 for rate independent material.
Default = 0

The FRCT keyword and parameters are optional. If specified they request a rate independent fracture energy-based softening. They preclude the specification of rate effects.

FRCT= (C) Characters FRCT (see Notes 1 and 2 below)

epsoft= (F) Axial strain at which softening begins in unixial stress.

asft= (F) Coefficient *a* in the softening function shown below. (Currently set to 1.0 by code regardless of input).

bsft= (F) Coefficient *b* in the softening function shown below.

The SFTN keyword and parameters are optional. If specified they request a strain-based softening. Rate effects may be included.

SFTN= (C) Characters SFTN .

epsoft= (F) Axial strain at which softening begins in unixial stress.

asft= (F) Coefficient *a* in the softening function shown below.

bsft= (F) Coefficient *b* in the softening function shown below.

decay= (F) Exponent governing decay of rate effects with softening. Set to zero for no decay of rate effects. Nearest integer value will be used. 0≤*decay*≤10. Default = 0.

edot, sig= (F) Pairs of plastic strain rate vs.. axial stress in uniax tension. These pairs all go on one line. Use the continuation character "&" as necessary.

Note: The softening function used is:

$$\sigma_1=\sigma_{ult}\left[\frac{\varepsilon^{psft}}{\varepsilon^{p}}(1-a)+a\exp\left(-b\left(\varepsilon^{p}-\varepsilon^{psft}\right)\right)\right] \quad \text{where,} \quad \varepsilon^{psft}=\varepsilon^{sft}-\frac{\sigma_{ult}}{E}$$

and σ_{ult}=min(*sigultimate*, *sigyield*+*hardng***epsoft*), *a*=*asoft*, *b*=*bsoft*, σ_1=axial stress, E = *emodulus*, ε^{sft} = *epsoft*, and ε^{p} = plastic axial strain. The functional coefficients *a* and *b* are fit to the axial stress vs.. plastic strain behavior in uniaxial stress tests.

2. Entering FRCT triggers a fracture energy-based softening algorithm for which the stress-displacement response is independent of discretization. For this case, rate effects are not included. Softening follows the functional form of note 1 with *a* set to 1.0 and plastic strain replaced by displacement across an infinitesimal band. For this case, the expended fraction of fracture energy is stored in the VDMG array.

3. The equivalent plastic strain (stored in the EL array) is the accumulation of $d\bar{\varepsilon}$

$$d\bar{\varepsilon}=\sqrt{\frac{2de_{ij}^{p}de_{ij}^{p}}{3}}$$

PROP (properties - PLIH) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for Piecewise Linear Isotropic hardening materials: *mattype* = PLIH.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** PLIH subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad sigyield gagelen curvnam ratenam ultrat dpflag beta c1 c2 epsf*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

sigyield = (F) Yield stress in uniaxial stress test. No default.

gagelen= (F) Gage length of test specimen. No Default.

curvnam = (C) Name of previously defined data array containing, engineering strain (column 1) vs.. engineering stress (column 2) for tension test on a specimen with length *gagelen*. See note 1. No default.

ratenam = (C) Name of previously defined data array containing strain rate (column 1) vs.. dynamic enhancement factor at yield (column 2) for tension test on specimen with length *gagelen*. Set to "0" for no rate enhancement (static). See note 2. No default.

ultrat = (F) Ratio of rate enhancement at ultimate to that at yield. *Ultrat* = $(\sigma_{ud}-\sigma_u)/(\sigma_{yd}-\sigma_y)$. Default=1. See note 3.

dpflag= (C) Drucker Prager flag. See Note 4. Default = ON.

PCOM SCOM input parameters

beta= (F) Drucker Prager position coefficient. The initial DP surface is located at beta * the ultimate radius of the Von Mises surface. Default = 2.

C1= (F) Coefficient of accumulated plastic strain in DP hardening/softening variable kappa. Default = -E/200.

C2= (F) Coefficient of volumetric plastic strain in DP hardening/softening variable kappa. C2=0 is recommended for hardening. Default = C1.

epsf= (F) The engineering failures strain for hardening DP surfaces (exact only for C2=0). Default = no failure.

Notes: 1. The curve can contain up to 20 points. The input strain values must be monotonically increasing. Ultimate stress is taken from the second to last point. Failure occurs at the last point. Softening is permitted only on the last segment. The last segment is used to compute fracture energy.

2. Enter up to 20 pairs of strain rate vs.. dynamic rate enhancement factor.
3. The rate enhancement decays as a function of equivalent plastic strain.
4. A Drucker-Prager surface is used to approximate the necking phenomenon within a band that is small compared to element size. *Dpflag* can be used to deactivate this feature.
5. Post-peak softening occurs only for equivalent plastic strains accumulated in the presence of tensile stresses. The expended fraction of post-peak plastic strain is stored in the VDMG array.
6. The (total) equivalent plastic strain (stored in the EL array) is the accumulation of $d\bar{\varepsilon}$

$$d\bar{\varepsilon} = \sqrt{\frac{2de^p_{ij}de^p_{ij}}{3}}$$

7. The post-peak softening slope is set to E/200. The strain to failure is adjusted based on element size to preserve fracture energy. At failure, the stress is reduced to zero over a strain range of 10 percent of ultimate.
8. The position of the DP surface is governed by Kappa = beta*vmrult + C1*epsacc + C2*epsvol. Where vmrult is the Von Mises stress at ultimate.

PROP (properties — SFT1) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* =SFT1.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** SFT1 subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype p1 p2 p3 p4 p5 p6 xio ptrns
pcut anon decay cr tcof cw cd esrat dmg0 tenhrd cmphrd
xint elstrt geop scof cd1 timfac npairs beta evbeg evend gage sigfac lngfac bexp
edot(1) sig(1) edot(2) sig(2) ...edot(npairs) sig(npairs)

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0

rltype= (F) Option flag. Either 6.0 or 7.0. See note 4. No default.

p1 = (F) Parameter *f'c* if rltype = 6, *c0* if rltype = 7. No default.
p2 = (F) Parameter *az* if rltype = 6, *c1* if rltype = 7.
p3 = (F) Parameter *au* if rltype = 6, *c2* if rltype = 7.
p4 = (F) Parameter *xi* if rltype = 6, *d0* if rltype = 7.
p5 = (F) Parameter *r1* if rltype = 6, *d1* if rltype = 7.
p6 = (F) Parameter *r2* if rltype = 6, *d2* if rltype = 7.

xio = (F) Pressure where failure surface is equal to zero. Default internally computed. See Note 5.

ptrns = (F) Transition point along pressure axis between scaled failure surface and unscaled failure surface. Default= *f'c*/3.

See Note 5.

pcut = (F) Pressure cutoff.

anon = (F) Volumetric nonassociativity ($0 \leq anon \leq 1$) Default=0.5.

decay= (F) Decay of rate effects. Default = 1.

cr = (F) Cap shape.

tcof = (F) Tension softening.

cw = (F) Cap-hardening parameter.

cd = (F) Cap-hardening parameter.

dmg0 = (F) Initial damage. Default=0.0. See Note 6.

tenhrd = (F) Rate coefficient for ductile damage in tension. Default=0.0. See Note 6.

cmphrd = (F) Rate coefficient for ductile damage in compression. Default=0.0. See Note 6.

xint = (F) Initial cap position.

elstrt = (F) Always use internal default.

geop = (F) Geostatic pressure.

scof = (F) Unconfined compression softening.
cd1 = (F) Cap-hardening parameter.

timfac = (F) Conversion from time units to msec. Default=1.

npairs= (F) Number of points ($\leq$ 20) defining rate dependent behavior in uniaxial compression. Default available. Enter 0 to disable rate effects.

beta= (F) Cap rate Coefficient. Default = 0

evbe.g., evend= (F) Associativity drops from *anon* to zero as volumetric plastic strain increases from *evbg* to *even*. Default=(0.08,0.11)

gage= (F) Element size for which properites are specified. Default=4.0 inch. See note 1 below.

sigfac= (F) Conversion from ksi to stress units. Default = 1.0.

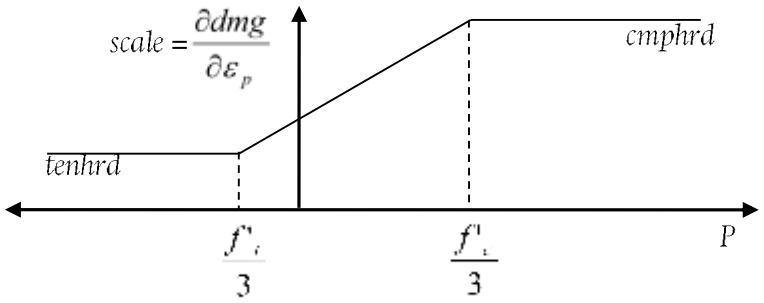
lngfac= (F) Conversion from inches to length units. Default=1.0

bexp= (F) Cap rate exponent. Default = 0.

edot, sig= (F) Pairs of plastic strain rate vs.. axial stress in unconfined compression. These pairs all go on one line.

MATR-PROP (SFT1)

Notes:
1. The tension softening parameter is adjusted based on element size to maintain approximately constant fracture energy when a nonzero size is entered for gage. No adjustmment is made when gage = 0.
2. The timfac, lngfac, sigfac are needed to compute default values based on CEB.
3. For this model, the cap position is stored in the EL array, the fraction of softening is stored in VDMG, and the accumulated volumetric plastic strain increments are stored in EVBR.
4. Entering -6.0 or -7.0 triggers a recompactible tension treatment.
5. Defining parameters *xio* and *ptrns* changes the shape of the failure surface for tensile pressures. The failure surface between *xio* and *ptrns* becomes $f(p) = A - C\,e^{B\,p}$ where, A, C, and B are internally computed to force the function to be zero at $p = xio$ and to match the slope of the failure surface at *ptrns*.
6. A more ductile response can be achieved in the model (e.g., for fiber-reinforced concrete) in tension and/or compression by defining parameters *dmg0*, *tenhrd* and *cmphrd*. Entering a negative value for *dmg0* causes softening to be delayed when the stress state reaches the failure surface until damage is greater than zero. *tenhrd* and *cmphrd* control the rate of growth of damage for damage less than zero according to $dmg^{n+1} = dmg^{n} + scale * devc$ where, *devc* is function of the equivalent plastic and volumetric plastic strain increments and *scale* is a function of pressure as shown.



PROP (properties - SFT2) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* =SFT2.

Use Is: **SFT2 not maintained. It has been deactivated.**

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** SFT2 subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad*
rltype p1 p2 p3 p4 p5 p6
pcut anon decay cr tcof cw cd
xint elstrt geop scof cd1 ho bwdth
ah bh ch timfac npairs
edot(1) sig(1) edot(2) sig(2) ...edot(npairs) sig(npairs)

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

rltype= (F) Option flag. Either 6.0 or 7.0. No default.

p1 = (F) Parameter *f'c* if rltype = 6, *c0* if rltype = 7. No default.
p2 = (F) Parameter *az* if rltype = 6, *c1* if rltype = 7.
p3 = (F) Parameter *au* if rltype = 6, *c2* if rltype = 7.
p4 = (F) Parameter *xi* if rltype = 6, *d0* if rltype = 7.
p5 = (F) Parameter *r1* if rltype = 6, *d1* if rltype = 7.
p6 = (F) Parameter *r2* if rltype = 6, *d2* if rltype = 7.
p7 = (F) Parameter not used.

anon = (F) Volumetric nonassociativity ($0 \leq anon \leq 1$). No default.

p8 = (F) Parameter not used.

cr = (F) Cap shape.

tcof = (F) Tension softening. No default.

cw = (F) Cap-hardening parameter.

cd = (F) Cap-hardening parameter.

xint = (F) Initial cap position.

elstrt = (F) Always use internal default.

geop = (F) Geostatic pressure.

scof = (F) Unconfined compression softening. No default.

cd1 = (F) Cap-hardening parameter.

ho = (F) Initial yield surface shift. Default = 0.5

bwdth = (F) Crack band width.

ah = (F) Yield surface-hardening parameter.

bh = (F) Yield surface-hardening parameter.

ch = (F) Yield surface-hardening parameter.

timfac = (F) Conversion from msec to time units. No default.

npairs= (F) Number of points ($\leq$ 10) defining rate dependent behavior in uniaxial compression. Default available.

edot, sig= (F) Pairs of plastic strain rate vs. axial stress in unconfined compression. These pairs all go on one line.

PROP (properties - EOS) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = EOS, nonlinear equation of state model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** EOS subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad eostype gamma energy*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

eostype = (F) Type of eos model. 1.0 = ideal gas, 2.0 = air, 3.0 = water. No default.

gamma= (F) Gamma for material. Default = 2.

energy= (F) Initial internal energy per unit mass. Default = 1.

PROP (properties - TISU) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = TISU, "B / A" tissue model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** TISU subcommand.

PCOM SCOM input parameters

PROP *matname density bulk shear hourglas damplinear dampquad bovera pmax pcut*

matname = (C) Material name. No default.

density = (F) Mass density. No default.

bulk = (F) Bulk modulus. No default.

shear = (F) Shear modulus. No default.

hourglas = (F) Hourglass suppression coefficient. Default = .01

damplinear = (F) Linear coefficient for artificial viscosity. Default = 0.0

dampquad = (F) Quadratic coefficient for artificial viscosity. Default = 0.0

bovera= (F) B/A. Default = 0.

pmax= (F) Estimate of maximum pressure in calculation. Input a negative number since compresssion is negative by convention. Default = 0.

pcut= (F) Pressure cutoff. Default = 0.

Notes: 1. This model is implements a quadratic pressure / volume relation:
2. For stability, an upper bound is needed on the bulk modulus. This is obtained as the tangent modulus at pmax. Linear response is implemented for higher (negative) pressures.
3. Pcut is an irreversible pressure cutoff. For reversible behavior, set pcut to a large positive number.
4. This model automatically calculates pressures without calculating stresses. The **CALC STRS** subcommand should be used to activate stresses if they are desired.

PROP (properties -YEOH) Subcommand
MATR (material) Input Group **MATR-PROP**

Purpose: To define the constitutive model parameters for materials with *mattype* = YEOH, hyperelastic model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow a **TYPE** YEOH subcommand.

PCOM	SCOM	input parameters
PROP		*matname density bulk shear hourglas damplinear dampquad c1 c2 c3 d1 d2 d3*

c1,c2,c3 = (F) Coefficients of distortional strain energy terms. Default = 0.

d1,d2,d3 = (F) Coefficients of volumetric strain energy terms. Default = 0.

Notes: 1. This hyperelastic model implements Yeoh's reduced polynomial form for the strain energy. It is valid up to very large (100's of percent) strain. See for example Chen, J.S. and C. Pan, "A Pressure Projection Method for Nearly Incompressible Rubber Hyperelasticity, Part I: Theory," *ASME Jnl Applied Mechanics*, Vol 63, 1996, or Yeoh, O.H. "Some Forms of the Strain Energy Function for Rubber," *Rubber Chemistry and Technoplogy*, Vol. 66, pp. 754-771, 1993. The Strain Energy is given as:

$$W = c_1(\bar{I}_1 - 3) + c_2(\bar{I}_1 - 3)^2 + c_3(\bar{I}_1 - 3)^3 + d_1(J-1)^2 + d_2(J-1)^4 + d_3(J-1)^6$$

where J is the determinant of the deformation gradient tensor, and $\bar{I}_1$ is the first invariant of the Green deformation tensor, normalized to remove volumetric components..

2. The 6 coefficients should be chosen to match lab data for a given material. The low-strain shear modulus is equal to 2** c1*, the low-strain bulk modulus is equal to 2**d1*. For rubberlike materials, c2 is negative and approximately one order of magnitude less than c1. c3 is positive and approximately 2 orders of magnitude less than c1.
3. This model is implemented for continuum and shell elements.
4. *bulk & shear* are used only for computing a stable timestep and hourglass resistance. They should typically be set to 2**d1* and 2**c1* respectively.
5. The quantity $\sqrt{\bar{I}_1 - 3}$ is stored in the EL array for continuum elements and shpl(1,j) for shell elements respectively. This provides a convenient measure of distortion and may be used as an erosion criterion.

HRGL (hourglass) Subcommand
MATR (material) Input Group **MATR-HRGL**

Purpose: To allow a user to activate or deactivate the damaged hourglass stiffness option for material types SFT1 and SFT2 or to select assumed strain type hourglass supression for linear elastic materials.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand which defined the selected material properties set.

PCOM SCOM input parameters

HRGL *matname keyword value*

matname = (C) Name of the material whose hourglass stiffness will be damaged during a calculation.
No default.

keyword = (C) Set to NODAMG to deactivate hourglass damage based on the constitutive models damage parameter (SFT1 and SFT2 materials only). Bt default, the hourglass stiffness for elements assigned SFT1 and SFT2 materials is linked to the value of VDMG for the element.

Set to FB to select Flanagan-Belytchko perturbation type hourglass supression (default).

Set to BB to select Belytchko-Bindeman type assumed strain hourglass supression (see note 1).

Set to VISC to add viscous hourglass suppression.

value = (F) Hourglass suppression coefficient for BB or FB. Default = 0.01 for FB ; 1.0 for BB.
Viscous hourglass coefficient to be used alone, or with FB stiffness control. Sqrt (stiffness coefficient) approximates critical damping of hourglass modes. Default = 0.

Notes: 1. The BB type hourglass supression is slightly slower than FB, but dramatically more accuracte for linear elastic materials. For anisotropic materials, an isotropic approximation is used. For nonlinear materials, BB is not much better than FB, and a small hourglass constant should be used. BB type hourglass supression provides no hourglass resistance for fluids because it is proportional to the shear modulus.

2. Hourglass damage is on by default for SFT1 materials. Keyword DAMG keeps it on, keyword NODAMG turns it off.

3. Combined Stiffness+viscous hourglass suppression is quite effective. See: Daniel, W. and T. Belytchko, "Suppression of spurious intermediate frequency modes in under-integrated elements by combined stiffness/viscous stabilization," Int. Jnl. Num. Meth. Engr., Vol 64, June 2005.

4. The viscous stabilization force is currently restricted to be no larger than the elastic increment (when non-zero) to avoid stability issues that sometimes appear.

IMPD (impedance) Subcommand
MATR (material) Input Group **MATR-IMPD**

Purpose: To specify a material's wave speed values to be used for the impedance boundary condition (absorbing boundary).

Use Is: Optional. If not input, elastic wave speeds are assumed.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

IMPD *matname cp cs*

matname = (C) Material name.
No default.

cp = (F) Dilatational wave speed.
Default = elastic dilatational wave speed.

cs = (F) Shear wave speed.
Default = elastic shear wave speed.

DAMP (damping) Subcommand
MATR (material) Input Group **MATR-DAMP**

Purpose: To specify an estimate of a material's peak volumetric strain rate in order to compute appropriate and stable time steps for the model when quadratic artificial viscosity is requested (i.e., *dampquad* ≥ 0.0).

Use Is: Optional. If not input, a volumetric strain rate of 0.0 is assumed for the material.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM	SCOM	input parameters
	DAMP	*matname volrate*

matname = (C) Material name.
No default.

volrate = (F) Volumetric strain rate.
Default = 0.0

Notes: 1. The volumetric strain rate for each element is stored in the VOLR(i,j,k) data array if the **CALC VOLR** command was input or if artificial viscosity is specified for any input material type. Time histories of the volumetric strain rate for any element can be retrieved using **POUT** or **OUTP** commands if the volumetric strain rate is unknown.

2. A value for *volrate* is required only if strain rates are great enough to cause stability problems in the model. The volumetric strain rate may be estimated as: $\frac{\Delta p}{\rho c d}$ where Δp is the stress jump at the shock front, is the density, d is the element dimension and c is the secant wavespeed corresponding to Δp.

WVSP (wave speed) Subcommand
MATR (material) Input Group **MATR-WVSP**

Purpose: To switch the material input mode from specifying the basic stiffness characteristics of a material in terms of its stiffness moduli to specifying them in terms of its dilatational and shear wave speeds. If switched to wave speed input, the input parameters *bulk* and *shear* on the **PROP** subcommand are interpreted as actually being the dilatational and shear wave speeds, respectively, of the material.

Use Is: Optional. If not input, the standard approach of inputting the bulk and shear moduli for a material is assumed.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must precede the **PROP** subcommands that define the material properties in the requested manner.

PCOM	SCOM	input parameters
	WVSP	*option*

option = (C) Chosen option. Any of:
ON = wave speed input
OFF = standard stiffness input

Default = ON

Note: 1 The relationships between the stiffness moduli of a material and its elastic wave speeds are:

$$c_p = \sqrt{\frac{K + (4G/3)}{\rho}}$$

and

$$c_s = \sqrt{\frac{G}{\rho}}$$

where K and G are the bulk and shear moduli, ρ is the mass density, and c_p and c_s are the dilatational and shear wave speeds of the material.

VDMP (viscous damping) Subcommand
MATR (material) Input Group **MATR-VDMP**

Purpose: To specify viscous damping parameters for a linear elastic material to convert it to a viscoelastic material with the specified damping. The form of the damping specification may be a Q value, a fraction of critical damping, or a db/distance at a specified frequency.

Use Is: Optional. Input only for linear elastic materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

VDMP *matname centfreq option dilatation shear frequency expv distance exps*

matname = (C) Material name. Must be a linear elastic material. No default.

centfreq= (F) The center frequency (cycles/unit time) at which the viscoelastic damping model matches the damping vs. frequency curve specified by the following parameters. (see Note 1). No default.

The following parameters define desired damping vs frequency

option = (C) Damping specification option. ANY of: Q, DB or CRIT. Set to Q if specifying damping loss as a Q value. Set to DB if specifying damping loss as a db/distance at a specified frequency. Set to CRIT if specifying damping loss as a fraction of critical damping. No default.

dilatation = (F) Damping for a dilational wave at *frequency*. Q value if *option*=Q. DB/distance if *option*=DB. Fraction of critical damping if *option* = CRIT. See note 3 below. No default.

shear = (F) Damping for a shear wave at *frequency*. Q value if *option*=Q. DB/distance if *option*=DB. Fraction of critical damping if *option* = CRIT. Default,=0.0, no shear damping. See note 3 below.

frequency = (F) Frequency (cycles/unit time) at which the values *dilatation* and *shear* are specified. Default=*centfreq*.

expv= (F) Exponent in a power law representation of the material's damping. Default = 0

distance = (F) Input only for *option*=DB. The distance over which the damping ocurrs in model units at *frequency*. Default = 1.0

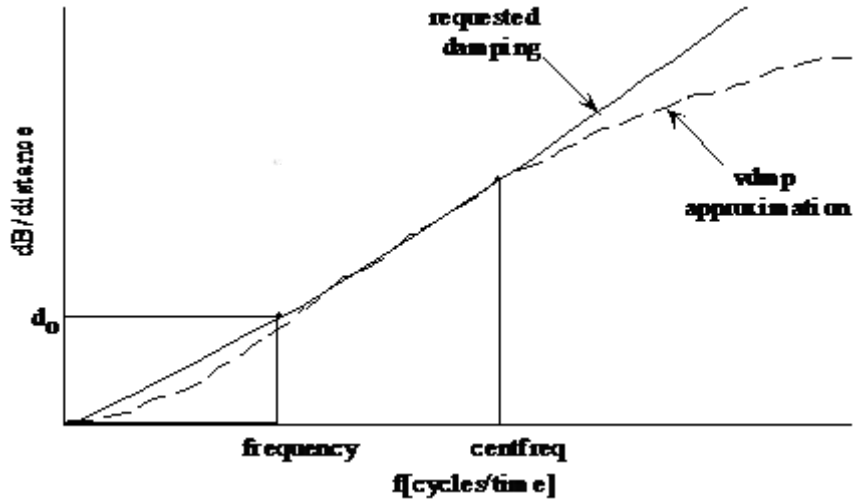
exps = (F) Exponent in a power law representation of the material's shear damping. Default = *expv*

Notes: 1. The standard solid viscoelastic model produces frequency dependent damping. The peak damping occurs at *centfreq* Damping is lower above and below this frequency. Damping is within about 80% of peak in the range .5**centfreq* to 2.**centfreq* The peak damping value is defined by the damping parameters input.

2. This damping model uses an initially stiffer modulus than the elastic stiffness, and relaxes to a lower stiffness over a relaxation time. This results in somewhat smaller time steps due to the faster initial wave speed of the material.
3. The damping functions independently on the volumetric and deviatoric strain states. A dilational wave includes both types of strain components, whereas a shear wave includes only a deviatoric strain component. This places some restrictions on the relative damping allowed for the two wave types when requested shear damping is much larger than requested dilatational damping. If the requested values result in a negative volumetric damping, the volumetric damping is set to zero and a warning is printed. In this case, the shear damping is correct, but the dilatational damping is too large.
4. The current initialization procedure is improved relative to that in earlier versions of PZFlex. The **OLD** primary command may be used to invoke the old initialization procedure for backward compatibility.
5. The dB is a measure of power loss. For 1D wave propagation, if A1 denotes the incident wave amplitude and A2 the amplitude after propagating *distance*, then
dB = 20 $\log_{10}$ (A1/A2).
6. A power-law representation of the power loss is used to facilitate damping specification:

$$dB/\,distance = d0\left(\frac{f}{frequency}\right)^{exponent}$$. Typically $1 \le exponent \le 2$.

Here, *d*0 is either *shear* or *dilatation* for s or p waves, respectively. The viscoelastic damping model matches the specified power-law damping curve at *centfreq*.



SDMP (stiffness damping) Subcommand
MATR (material) Input Group **MATR-SDMP**

Purpose: To specify stiffness-proportional damping parameters for a material. May be used for linear elastic or linear anisotropic materials.

Use Is: Optional. Precludes the use of artificial viscosity in the model. Not valid for viscoelastic materials. Not implemented for large deformation. An isotropic approximation is used for anisotropic materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

SDMP *matname centfreq option dilatation shear frequency expv distance exps*

matname = (C) Material name. No default.
centfreq = (F) Frequency (cycles/unit time) at which the stiffness proportional damping model matches the damping vs.. frequency curve specified by the following parameters. No default.

The following parameters define desired damping vs. frequency

option = (C) Damping specification option. ANY of: Q, DB or CRIT. Set to Q if specifying damping loss as a Q value. Set to DB if specifying damping loss as a db/distance at a specified frequency. Set to CRIT if specifying damping loss as a fraction of critical damping. No default.

dilatation = (F) Damping for a dilatational wave at *frequency*. Q value if *option* = *Q*. DB/distance if *option* = *DB*. Fraction of critical damping if *option* = CRIT. No default.

shear = (F) Damping for a shear wave at *frequency*. Q value if *option* = *Q*. DB/distance if *option* = *DB*. Fraction of critical damping if *option* = CRIT. Default = 0.0 (see Note 6).

frequency = (F) Frequency (cycles/unit time) at which the values *shear* and *dilatation* are specified. Default = *centfreq*.

expv = (F) Exponent in a power-law representation of the material's damping . Default = 0.0.

distance = (F) Input only for *option* = *DB*. The distance over which the damping occurs in model units at *frequency*.
Default = 1.0.

exps = (F) Exponent in a power law representation of the material's shear damping. Default = *expv*

Notes: 1. The stiffness-proportional model produces frequency dependent damping. The amplitude decay per unit distance increases approximately linearly with frequency.

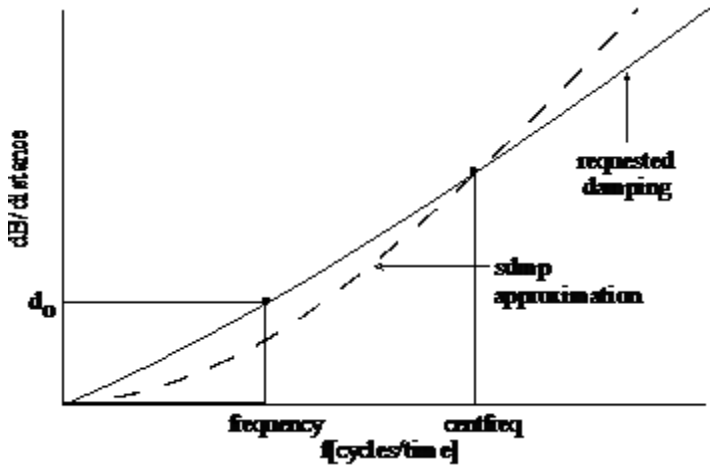
2. The damping functions independently on the volumetric and deviatoric strain rates. A dilatational wave includes both types of strain components while a shear wave includes only a deviatoric strain component. This places a lower limit on the amount of dilatational wave damping relative to the shear wave damping (to maintain a non-negative volumetric damping coefficient). If the input results in a negative volumetric damping coefficient, it will be set =0.0 and a warning message will be printed.

3. An isotropic approximation is used for anisotropic elastic models.

4. The dB is a measure of power loss. For 1D wave propagation, if A1 denotes the incident wave amplitude and A2 the amplitude after propagating *distance*, then
dB = 20 $\log_{10}$ (A1/A2).

5. A power-law representation of the power loss is used to facilitate damping specification:
$dB/\ distance = d0\left(\frac{f}{frequency}\right)^{exponent}$. Typically $1 \le exponent \le 2$.
Here, *d*0 is either *shear* or *dilatation* for s or p waves, respectively. The stiffness proportional damping model will match the specified power-law damping curve at *centfreq*.

6. For acoustic materials (shear modulus =0), the damping is taken from the dilatational specification. In this case *shear* is a number between 0 and 1 (default = 0) defining how the damping is split between shear and volumetric strain components.
$shear = \frac{\beta}{\beta + \frac{4}{3}\mu}$ where β and μ are the volumetric & shear viscosities respectively.
For *shear* = 0, all damping is due to shear viscosity, for *shear* = 1, all damping is due to volumetric viscosity.



MDMP (mass damping) Subcommand
MATR (material) Input Group **MATR-MDMP**

Purpose: To specify "mass-proportional" damping parameters for a material. May be used for any material. The damping values are based on elastic moduli for nonlinear materials.

Use Is: Optional. Precludes the use of dynamic relaxation in the model. Not valid for viscoelastic materials. An isotropic approximation is used for anisotropic materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

MDMP *matname centfreq option damping wavetype frequency expv distance exps*

matname = (C) Material name. No default.

centfreq = (F) Frequency (cycles/unit time) at which the mass proportional damping model matches the damping vs.. frequency curve specified by the following parameters. No default.

The following parameters define desired damping vs. frequency

option = (C) Damping specification option. ANY of: Q, DB or CRIT. Set to Q if specifying damping loss as a Q value. Set to DB if specifying damping loss as a db/distance at a specified frequency. Set to CRIT if specifying damping loss as a fraction of critical damping. No default.

damping= (F) Damping for a *wavetype* wave at *frequency*. Q value if *option* = *Q*. DB/distance if *option* = *DB*. Fraction of critical damping if *option* = CRIT. No default.

wavetype= (F) Type of wave for which damping is specified. Either DIL or SHER Default = DIL (see Note 6).

frequency = (F) Frequency (cycles/unit time) at which the value *damping* is specified. Default = *centfreq*.

expv = (F) Exponent in a power-law representation of the material's damping. Default = 0.0.

distance = (F) Input only for *option* = *DB*. The distance over which the damping occurs in model units at *frequency*.
Default = 1.0.

exps = (F) Exponent in a power law representation of the material's shear damping. Default = *expv*

Notes: 1. The mass-proportional model produces damping such that the amplitude decay per unit distance is approximately constant with frequency.

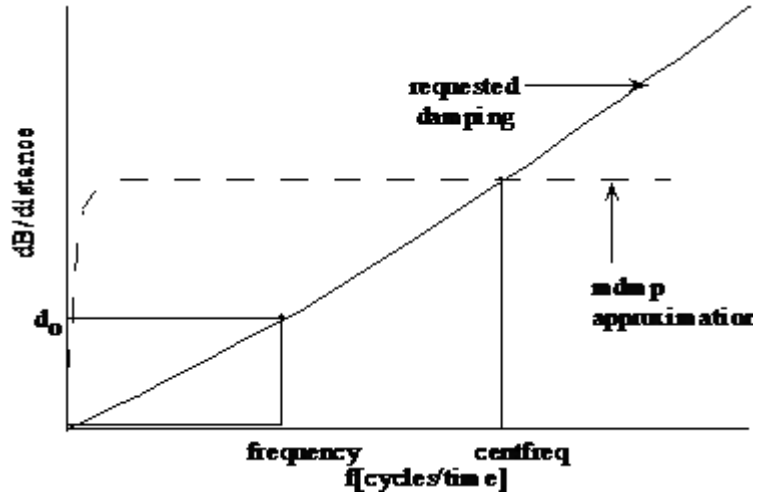
2. The damping is applied to the velocities rather than to the strain rates (as in VDMP or SDMP models). This may be envisioned as independent dashpots connected between each node and ground. The damping is specified for one wave type and this produces the same Q and % critical damping values for the other wave type. dB loss values are printed for both wave types, but the user can control only one of them.

3. An isotropic approximation is used for anisotropic elastic models.

4. The dB is a measure of power loss. For 1D wave propagation, if A1 denotes the incident wave amplitude and A2 the amplitude after propagating *distance*, then
dB = 20 $\log_{10}$ (A1/A2).

5. A power-law representation of the power loss is used to facilitate damping specification:
$dB/\ distance = d0\left(\frac{f}{frequency}\right)^{exponent}$. Typically $1 \le exponent \le 2$.
Here, *d*0 is *damping*. The mass proportional damping model will match the specified power-law damping curve at *centfreq*.

6. For acoustic materials (shear modulus =0), the damping must be specified for a dilataional wave.



RDMP (Rayleigh damping) Subcommand
MATR (material) Input Group **MATR-RDMP**

Purpose: To specify Rayleigh damping parameters for a material. May be used for linear elastic or linear anisotropic materials.

Use Is: Optional. Precludes the use of artificial viscosity in the model. Not valid for viscoelastic materials. Not implemented for large deformation. An isotropic approximation is used for anisotropic materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

RDMP *matname centfreq option dilatation shear frequency expv distance exps*

matname = (C) Material name. No default.

centfreq = (F) Frequency (cycles/unit time) at which the Rayleigh damping model matches the damping vs.. frequency curve specified by the following parameters. No default.

The following parameters define desired damping vs. frequency

option = (C) Damping specification option. ANY of: Q, DB or CRIT. Set to Q if specifying damping loss as a Q value. Set to DB if specifying damping loss as a db/distance at a specified frequency. Set to CRIT if specifying damping loss as a fraction of critical damping. No default.

dilatation = (F) Damping for a dilatational wave at *frequency*. Q value if *option* = *Q*. DB/distance if *option* = *DB*. Fraction of critical damping if *option* = CRIT. No default.

shear = (F) Damping for a shear wave at *frequency*. Q value if *option* = *Q*. DB/distance if *option* = *DB*. Fraction of critical damping if *option* = CRIT. Default = 0.0 (see Note 6).

frequency = (F) Frequency (cycles/unit time) at which the values *shear* and *dilatation* are specified. Default = *centfreq*.

expv = (F) Exponent in a power-law representation of the material's damping . Default = 0.0.

distance = (F) Input only for *option* = *DB*. The distance over which the damping occurs in model units at *frequency*.
Default = 1.0.

exps = (F) Exponent in a power law representation of the material's shear damping. Default = *expv*

Notes: 1. The Rayleigh damping model produces frequency dependent damping. The amplitude decay per unit distance increases approximately linearly with frequency.

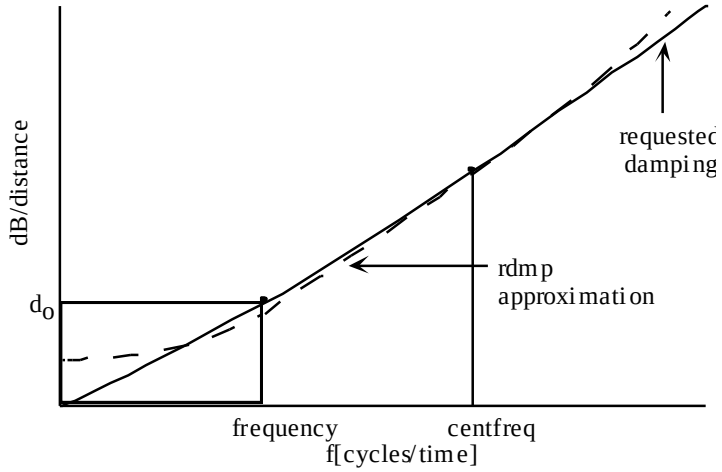
2. The damping functions independently on the volumetric and deviatoric strain rates. A dilatational wave includes both types of strain components while a shear wave includes only a deviatoric strain component. This places a lower limit on the amount of dilatational wave damping relative to the shear wave damping (to maintain a non negative volumetric damping coefficient). If the input results in a negative volumetric damping coefficient, it will be set =0.0 and a warning message will be printed.

3. An isotropic approximation is used for anisotropic elastic models.

4. The dB is a measure of power loss. For 1D wave propagation, if A1 denotes the incident wave amplitude and A2 the amplitude after propagating *distance*, then
dB = 20 $\log_{10}$ (A1/A2).

5. A power-law representation of the power loss is used to facilitate damping specification:
$dB/\ distance = d0\left(\frac{f}{frequency}\right)^{exponent}$. Typically $1 \le exponent \le 2$.
Here, *d*0 is either *shear* or *dilatation* for s or p waves, respectively. The stiffness proportional damping model will match the specified power-law damping curve at *centfreq*.

6. For acoustic materials (shear modulus =0), the damping is taken from the dilatational specification. In this case *shear* is a number between 0 and 1 (default = 0) defining how the damping is split between shear and volumetric strain components.
$shear = \frac{\beta}{\beta + \frac{4}{3}\mu}$ where β and μ are the volumetric & shear viscosities respectively.
For *shear* = 0, all damping is due to shear viscosity, for *shear* = 1, all damping is due to volumetric viscosity.



Notes: 7. Rayleigh damping combines the SDMP and MDMP models. For a given frequency range, it often provides a better approximation than either MDMP or SDMP.

8. This model matches the requested wavespeeds and damping values at *centfreq*, and minimizes the weighted error between requested and actual damping values over the range (*frqmin* ≤ *freq* ≤ *frqmax*). For *wtshear* = 0. the dilatational damping is optimized, neglecting the shear wave damping.. For *wtshear* = 1, the shear damping is optimized, neglecting the dilatational damping. For values between 0 and 1, the weighted error is minimized.

9. *Frqmin, frqmax* and *wtshear* can be set via the FRNG subcommand. If not set, they default to *frqmin* = 0.5**centfreq*, *frqmax* = 2.0**centfreq* and *wtshear* = 0.5.

SLOSS (Simplified Loss) Subcommand
MATR (Material) Input Group **MATR-SLOSS**

Purpose: To request that simplified estimates of material losses be computed for a material. This method just estimates losses as a factor multiplying energy — it does not introduce damping into the system.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MATR** Group: Must follow the **PROP** subcommand which defines the material properties.

PCOM SCOM input parameters

SLOSS *matname mechfactr dielfactr piezfactr*

matname = (C) Material name. No default.

mechfactr = (F) Mechanical loss coefficient. Default = 0.

dielfactr = (F) Dielectric loss coefficient. Default = 0.

piezfactr = (F) Coupling loss coefficient. Default = 0.

Notes:
1. Mechanical losses are computed as *mechfactr* * CLPE (See CALC PZENRG). These are stored in the SLSE array.
2. Dielectric Losses are computed as *dielfactr* * CLPD. (See CALC PZENRG) These are stored in the SLSD array. Coupling losses are computed as *piezfactr* * CLPP. (See CALC PZENRG) These are stored in the SLSP array

FRNG (frequency range) Subcommand
MATR (material) Input Group **MATR-FRNG**

Purpose: To specify a frequency range and weights for Rayleigh damping optimization.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MATR** Group: Must preceed any **RDMP** subcommands which to which it applies.

PCOM	SCOM	input parameters
	FRNG	*frqmin frqmax wtshear*

frqmin = (F) Lower bound of frequency range. Default = 0.

frqmax = (F) Upper bound of frequency range. Default = 0.

wtshear = (F) Shear wave weighting factor (0 ≤ *wtshear* ≤ 1). Default = 0.5

Notes: 1. If *frqmin* = *frqmax* = 0., the default frequency interval (0.5*centfreq*, 2.0*centfreq*) will be used.
2. The dilatational wave damping is weighted by (1.0 - *wtshear*).

PRNT (print) Subcommand
MATR (material) Input Group **MATR-PRNT**

Purpose: To request additional information about material models.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the subcommands which defines the material properties.

PCOM SCOM — input parameters

PRNT DAMP *matname frqmin frqmax frqinc (filename)*
PIEZ *matname (filename)*
STIF *matname (filename)*

option= (C) Print option. Any of: DAMP.

For option = DAMP

matname = (C) Material name. No default.

frqmin = (C) Lower bound of frequency range. Default = 0.

frqmax= (C) Upper bound of frequency range. Default = 0.

frqinc= (F) Frequency increment. Default = 0.

filename= (C) Filename (up to 80 characters) to which damping vs frequency information is to be written. The information will be written to the flxprt file if no filename is specified.

For option = PIEZ or STIF

The Stiffness or Piez coupling constants transformed to global coordinates are written to the specified filename. If no *filename* is specified, they are printed to the *flxprt* file. Currently, stiffness is only written for anisotropic elastic materials.

Note: The DAMP option produces a 5 column table consisting of: frequency, requested dB loss/unit distance for a dilatational wave, actual dB loss /unit distance for a dilatational wave, requested dB loss/unit distance for a shear wave and actual dB loss /unit distance for a shear wave. This information can be written to a file and plotted with Review. It is useful for quantifying how damping varies with frequency.

NWTN (Newtonian viscosity) Subcommand
MATR (material) Input Group **MATR-NWTN**

Purpose: To specify linear viscosity coefficients for a material.

Use Is: Optional. Precludes the use of artificial viscosity in the model. Not valid for viscoelastic materials. Not implemented for large deformation. An isotropic approximation is used for anisotropic materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM	SCOM	input parameters
	NWTN	*matname volumetric shear*

matname = (C) Material name. No default.

volumetric = (F) Volumetric viscosity coefficient. Default = 0.

shear = (F) Deviatoric viscosity coefficient. Default = 0.

Notes: 1. The Newtonian viscosity model produces frequency dependent damping. The amplitude decay per unit distance increases approximately linearly with frequency.
2. The damping functions independently on the volumetric and deviatoric strain rates.

AXIS Subcommand
MATR (Material) Input Group **MATR-AXIS**

Purpose: To specify the local coordinate system in which an anisotropic material's constitutive parameters have been specified. The material parameters will be automatically transformed into the global coordinate system for the model being used.

Use Is: Optional. If no axis has been specified, an anisotropic material's constitutive parameters are assumed to be specified in the global coordinate system.

Multiple Use: Allowed.

Order Dependence Within **MATR** Group: Must follow the **PROP** subcommand which defines the material properties. Also, the referenced axis must have been previously defined using the **AXIS** primary command prior to entering the **MATR** command group.

PCOM SCOM input parameters

AXIS *matname axisname*

or (see Note 4)

AXIS *matname axisname option thetabeg thetaend ntheta phibeg phiend nphi*

or (see Notes 4, 5)

AXIS *matname axisname* GNRL *v1x v1y v1z v2x v2y v2z &*
thetab thetaend ntheta phibeg phiend nphi

matname = (C) Material name. No default.

axisname = (C) The axis name defining the local coordinate system in which the constitutive parameters were specified.
Default = STND (i.e. standard global coordinates)

option = (C) Shortcut. Enter PLUSR for poling in the radial direction (i.e. the z' axis of the material coincides with the local radial direction). MINUSR for poling in the negative radial direction.

thetabeg, thetaend, ntheta = (C) *ntheta* orientations will be defined between *thetabeg*, *thetaend*. Theta should be input in degrees.

phibeg, phiend, nphi = (C) For spherical systems, *nphi* orientations will be defined between *phibeg, phiend*. phi should be input in degrees.

(continued next page)

MATR-AXIS

PCOM SCOM input parameters

vx1, vy1, vz1 = (C) The orientation from the material definition that maps to radial. (0,0,1) would produce the same result as PLUSR

vx1, vy1, vz1 = (C) The orientation from the material definition that maps to theta.

Notes:
1. This option is useful for orienting anisotropic materials in non-Cartesian directions.
2. This option provides a simple means of transforming textbook material constants to a model's global coordinate convention (if they differ). For example, if transversely isotropic material constants are known with z as the normal to the isotropic plane (the material's local system) and the user wants to build a model with x as the normal to the isotropic plane (the global system), then the textbook values can be directly input in the local system and an **AXIS** subcommand can be used to specify a previously defined local coordinate system corresponding to the materials local system. In this case, the **AXIS** command would have defined a local axis (x', y', z') where x' coincides with y, y' coincides with z, and z' coincides with x. The constitutive properties defined in the local system are automatically transformed into the global system for use by the model.
3. Elastic, dielectric (if any), thermomechanical couplin g(if any) and piezoelectric coupling (if any) constants are transformed.
5. For cylindrical or spherical systems, multiple materials with the same basic properties but different orientations may be automatically generated. These will be assigned names *matname:1*, *matname:2*, etc. As a shortcut, PLUSR & MINUSR map the z' direction of the defined material to radial in AXISNAME.
6. For more general orientations, **V1**defines the material direction that maps to radial, and **V2** is the material direction that maps to theta.

AFRC (absorption fraction) Subcommand
MATR (material) Input Group **MATR-AFRC**

Purpose: To specify that only a fraction of dissipated energy will be added to the loss array for this material. Typically used where a portion of attenuation is due to diffuse scattering and the rest is due to absorption. Only the portion due to absorption is accumulated in the loss array for subsequent thermal calculations.

Use Is: Optional. Not implemented for large deformation.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

AFRC *matname fractabsorb*

matname = (C) Material name. No default.

fractabsorb = (F) Fraction of calculated energy dissipation accumulated in the LOSS array. Default = 1.0.

PIEZ (piezoelectric) Subcommand
MATR (material) Input Group **MATR-PIEZ**

Purpose: To specify a material's piezoelectric constitutive parameters. These are the [3x6] array of piezoelectric coupling constants.

Use Is: Optional. Input only for piezoelectric materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

PIEZ *matname (option) irow1 jcol1 value1 irow2 jcol2 value2 ... &*
irown jcoln valuen

matname = (C) Material name.
No default.

option = (C) Optional input parameter defining whether the piezoelectric constants are stress or strain type.
Any of: STRS = stress constants (***e***)
STRN = strain constants (***d***)
If not input, stress constants are assumed.

irown, jcoln = (I) Row and column position in the [3x6] matrix of stress constants. Notation as per Auld below.
No default.

valuen = (F) Stress constant for the matrix ***e***(*irown,jcoln*) or strain constant for the matrix ***d***(*irown,jcoln*), depending on *option* above.
Default = 0.0.

Notes:
1. All stress constants are zero except those explicitly set with the **PIEZ** subcommand.
2. If the number of constants exceeds what may be input on one line, the user may use continuation lines by using the & continuation character. Only one **PIEZ** subcommand per material is permitted.
3. If STRN constants are input, d(i,j) are NOT the tensor values. See Auld, B.A., *Acoustic Fields and Waves in Solids*, R.E. Krieger Publishing Co., Malabar, Fl., 1990., p272.

ELEC (electric) Subcommand
MATR (material) Input Group **MATR-ELEC**

Purpose: To specify a material's dielectric (permittivity) constitutive parameters.

Use Is: Optional. Need be input only for materials within the electric window defined by the **PIEZ WNDO** command.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

ELEC *matname (option) epsilonx epsilony epsilonz*

matname = (C) Material name.
No default.

option = (C) Optional input parameter defining whether the epsilon values are defined at constant stress or at constant strain.
Any of: STRS = constant stress (ε^T)
STRN = constant strain(ε^S)
If not input, constant strain parameters are assumed.

epsilonx = (F) Dielectric constant in the local x'-direction of the material.
No default.

epsilony = (F) Dielectric constant in the local y'-direction of the material.
Default=*epsilonx*.

epsilonz = (F) Dielectric constant in the local z'-direction of the material.
Default=*epsilonx*.

Note: The dielectric constants specified are the principal values of . The local x', y', and z'-directions are assumed to be the material's principal directions. If the x'-, y'-, and z'-directions differ from the global x-, y- and z-directions of the model, use the **MATR AXIS** subcommand in conjunction with the **AXIS** command group to define a local coordinate system defining the material's principal axes relative to the global system.

THEX (thermomechanical coupling) Subcommand
MATR (material) Input Group **MATR-THEX**

Purpose: To specify a material's thermomechanical coupling parameters.

Use Is: Optional..

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

THEX *matname alphax alphay alphaz*

matname = (C) Material name. No default.

alphax = (F) Thermal expansion constant in the local x'-direction of the material. No default.

alphay = (F) Thermal expansion constant in the local y'-direction of the material. Default=*alphax*.

alphaz = (F) Thermal expansion constant in the local z'-direction of the material. Default=*alphax*.

Note: The coupling constants specified are the principal values of α. The local x', y', and z'-directions are assumed to be the material's principal directions. If the x'-, y'-, and z'-directions differ from the global x-, y-, and z-directions of the model, use the **MATR AXIS** subcommand in conjunction with the **AXIS** command group to define a local (Cartesian) coordinate system defining the material's principal axes relative to the global system.

STOR (store) Subcommand
MATR (material) Input Group **MATR-STOR**

Purpose: To give the user the option of storing independent elastic material constants for each element in the model. This differs from the default approach of using table lookups to get material properties which saves memory storage. The elastic constants may be taken from the **MATR** and **SITE** commands in the normal way of defining material properties for the model, or the user may completely supersede all **MATR** and **SITE** input by providing a user-generated set of material properties for the model.

Warning: This option is not general purpose and can be used only with a site model that is linear elastic. Also, the **CALC STRN** option and the **MODL** command may not be used with this option.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: None.

PCOM SCOM input parameters

STOR *option*

option = (C) Either BASE or USER.
Default = BASE

If *option* = BASE: the elastic constants are taken from the **MATR** and **SITE** command groups in the normal way of assigning material properties to grid elements.

If *option* = USER: the elastic wave speeds and mass density must be provided by the user for each element in the grid. Before the **PRCS** command, the user must **DATA IN** the dilatational wave speed, shear wave speed and mass density with data array names CP, CS and RHO, respectively.

PERM (magnetic permeability) Subcommand
MATR (material) Input Group **MATR-PERM**

Purpose: To specify a material's magnetic (permeability) constitutive parameters.

Use Is: Optional. Need be input only for materials within the magnetic window defined by the **MAGN WNDO** command.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

PERM *matname (option) permx permy permz*

matname = (C) Material name.
No default.

option = (C) Optional input parameter defining whether the permeability values are defined at constant stress or at constant strain.
Any of: STRS = constant stress (μ^T)
STRN = constant strain(μ^S)
If not input, constant strain parameters are assumed.

permx = (F) Permeability in the local x'-direction of the material.
No default.

permy = (F) Permeability in the local y'-direction of the material.
Default=*permx*.

permz = (F) Permeability in the local z'-direction of the material.
Default=*permx*.

Note: The permeabilities specified are the principal values of μ. The local x', y', and z'-directions are assumed to be the material's principal directions. If the x'-, y'-, and z'-directions differ from the global x-, y-, and z-directions of the model, use the **MATR AXIS** subcommand in conjunction with the **AXIS** command group to define a local coordinate system defining the material's principal axes relative to the global system.

MGST (magnetostrictive) Subcommand
MATR (material) Input Group **MATR-MGST**

Purpose: To specify a material's magnetostrictive constitutive parameters. These are the [3x6] array of magnetostrictive coupling constants.

Use Is: Optional. Input only for magnetostrictive materials.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM	SCOM	input parameters
	MGST	*matname (option) irow1 jcol1 value1 irow2 jcol2 value2 ... & irown jcoln valuen*

matname = (C) Material name.
No default.

option = (C) Optional input parameter defining whether the magnetostrictive constants are stress or strain type.
Any of: e = stress constants (***e***)
d = strain constants (***d***)
If not input, stress constants are assumed.

irown, jcoln = (I) Row and column position in the [3x6] matrix of stress constants. Notation as per Auld below.
No default.

valuen = (F) Stress constant for the matrix ***e***(*irown,jcoln*) or strain constant for the matrix ***d***(*irown,jcoln*), depending on *option* above.
Default = 0.0.

Notes:
1. All stress constants are zero except those explicitly set with the **MGST** subcommand.
2. If the number of constants exceeds what may be input on one line, the user may use continuation lines by using the "&" continuation character. Only one **MGST** subcommand per material is permitted.
3. If strain constants are input, d(i,j) are NOT the tensor values. See Auld, B.A., *Acoustic Fields and Waves in Solids*, R.E. Krieger Publishing Co., Malabar, Fl., 1990., p272.

ELST (electrostrictive) Subcommand
MATR (material) Input Group **MATR-ELST**

Purpose: To specify a material's electrostrictive constitutive parameters.

Use Is: Optional. Input only for electrostrictive materials.
Note: Electrostrictive coupling implemented only for 3D models.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM	SCOM	input parameters
	ELST	*matname psat kappa epsilon0 q111 q211*

matname = (C) Material name. No default.

psat = (F) Spontaneous Polarization. No Default

k= (F) k in Hom & Shankar's model
No default.

epsilon0 = (F) Permittivity of free space in problem units.
Default = 0.0.

q111,q122= (F) Coupling coefficients
Default = 0.0.

Notes: 1. See C.L. Hom and N. Shankar, "A Fully Coupled Constitutive Model for Electrostrictive Ceramic Materials," *Jnl of Intelligent Material Systems and Structures*, Vol. 5, pp795-801, 1994.
2. Only one **ELST** subcommand per material is permitted.

RATE Subcommand
MATR (material) Input Group **MATR-RATE**

Purpose: To specify viscoplastic rate dependence for a CAP or SFT1 type material.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM	SCOM	input parameters
	RATE	*matname beta expnt edot1 factor1 edot2 factor2 ....edotn factorn*

matname = (C) Material name. No default.

beta = (F) CAP enhancement factor. Default = 0.

expnt = (F) CAP enhancement exponent. Default = 0.

edotn, factorn = (F) Pairs of strain rate, dynamic enhancement factor defining the rate dependence of the material.

Notes:
1. For many geologic materials, the slope of the uniaxial compressibility curve appears to increase with increasing strain rate. *Beta, expnt* adjust the baseline rate enhancement on the cap to include this effect.
2. Pairs of strain rate vs.. dynamic enhancement factor in unconfined compression define the baseline viscoplastic rate enhancement.
3. For SFT1 materials, the rate enhancement may alternatively be prescribed on the **PROP** command.

THRM (thermal) Subcommand
MATR (material) Input Group **MATR-THRM**

Purpose: To specify thermal parameters for a material.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: Must follow the **PROP** subcommand that defines the material properties.

PCOM SCOM input parameters

THRM *matname C kx' ky' kz' hgls cb wb Tb*

or

THRM *matname* TABL *subdir*

matname = (C) Material name. No default.

C = (F) Specific heat of material. No default.

kx', ky', kz' = (F) Thermal conductivities in x', y', and z' directions. No default for *kx'*. Default = *kx'* for *ky'* and *kz'*.

hgls= (F) Hourglass suppression coefficient. If *hgls* < 0, exact integration will be used; if *hgls* = 0, single point integration without hourglass suppression will be used; and if *hgls* > 0, single point integration with hourglass suppression will be used Default = 1.0.

cb= (F) Specific heat of fluid flowing through material. Default = 0.

wb= (F) Fluid mass flow rate per unit volume. Default = 0.

Tb= (F) Temperature of fluid. Default = 0

subdir= (C) Name of subdirectory conataining TABL data. Default = ' ', main directory..

Notes:
1. *Cb, wb* and *Tb* are used to model perfusion in tissue.
2. The TABL keyword enables tabular input of specific heat and conductivity as a function of temperature and/or the VRAT array. For this option, the user must have created the arrays: c:matname, kx:matname, ky:matname (and kz:matname for 3D problems). These arrays are dimensioned (nt,nv) where nt is the number of termperatures and nv is the number of VRAT entries. If nt > 0, the corresponding temperature array tmpr:matname must also be entered. Likewise, if nv > 0, vrat:matname must be entered. Tmpr and vrat must be monotonically increasing. The tables are evaluated using bilinear interpolation. The tmpr array is internally created by PZFlex, though if initial temperatures are to be specified, it should be created (via a DATA OPEN subcommand) and initialized prior to the PRCS step. Tmpr is a nodal array with dimensions equal to the number of nodes in the thermal window. VRAT is an elemental array with dimensions equal to the number of elements in the thermal window. It is not generated by PZFlex and must be created by the user.

COPY Subcommand
MATR (Material) Input Group **MATR-COPY**

Purpose: To copy an entire material definition to a new material with a new name.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MATR** Group: Must follow the **PROP** subcommand for the *copyfrom* material.

PCOM	SCOM	input parameters
	COPY	*copyfrom copyto*

copyfrom = (C) A previously defined material property name to copy from.
No default.

copyto = (C) Unique property name for the new material.
No default.

Notes: 1. This command will copy a material definition in whatever state it exists at the time of copying including parameters set by the PROP subcommand and by any other subcommand that alters material properties such as HRGL, IMPD, THRM, etc. After copying the *copyto* material is independent of the *copyfrom* material.

CURV (Curve) Subcommand
MATR Input Group **MATR-CURV**

Purpose: To provide a simple way of inputting digitized x-y data into a user specified Data Manager array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MATR** Group: Must be input before it is referenced.

PCOM SCOM input parameters

CURV *dataname nvalues filename scalex scaley shiftx (option)*

dataname = (C) The Data Manager array name.
No default.

nvalues = (I) The number of input pairs.
Default: If *filename* is specified, the user may default *nvalues* and the program will read information from filename until the first blank line is encountered or the end-of-file is detecte.

filename = (C) Name of a file containing the values. Default is to read the x,y values immediately following the **MATR CURV** command from the Job Input File

scalex = (F) Scale factor to scale the x values by.
Default = 1.0

scaley = (F) Scale factor to scale the y values by.
Default = 1.0

shiftx = (F) Shift value to add to the x values in *filename*.
Default = 0.0

option = (F) Optional parameter to control checking for monotonically increasing x. If not input, the x information for this set must be monotonically increasing. If set to NCEQ, then duplicate x values are allowed but x values that are less than preceding x entries are not allowed. If set to NCLE, then x values are not checked for monotonic increasing values. Note that allowing non-monotonic data for a loading function may cause difficulties when interpolating the xhistory data for a value at a specified x.
No default

(**CURV** subcommand continued on next page)

(continued)

Notes: 1. The format of the data is: one pair (x, y) per record for *nvalues* records. The data must be in free-format text form.

Example 1:
```
MATR CURV  H1  3        /* read in time history H1 from Job Input File
    0.0        0.0      /* one data pair is specified on each line, time is input first
    1.0        10.0
    5.0        0.0
```

Example 2:
```
MATR CURV  H3  3  MY.HIST.FILE        /* read in time history H3 from another file

                  < this data must be placed in file: MY.HIST.FILE >
    0.0        0.0              /* one data pair is specified on each line, time is input first
    1.0        10.0
    5.0        0.0
```

PARM (parameters) Subcommand
MATR (material) Input Group **MATR-PARM**

Purpose: To control whether internal material property tables are prepared to support access by the Symbol language.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MATR** Group: None.

PCOM SCOM input parameters

PARM *store_option*

store_option = (C) Controls whether material parameter arrays are created to support the **SYMB** #GET MATPROP function. This function provides the Symbol language with access to all internal material parameters. Any of: YES or NO.
Default: YES

MBRN (membrane) Input Command

Purpose: To define a 4-noded membrane, plane stress element for 2D and 3D models. Restricted to small deformation analyses.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence:Must follow the **GRID** command and precede the **PRCS** command.

PCOM SCOM input parameters

MBRN

PROP	*membname rho emodulus poisson sigyield thickness tensyield hourglas*
ELEM	*membname ibegin iend jbegin jend kbegin kend*
SNGL	*membname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
PRNT	*option*
END	

Notes:
1. The material constitutive relations assume an elastic, perfectly plastic von Mises yield criterion.
2. The mass of membrane elements are lumped to the element's nodes and the stability characteristics of membrane elements are considered when computing zone time steps.
3. Stresses and strains for each membrane element are stored in the MBRD(i,j) data array. The i index of this array ranges from 1 to 6 where:

   i = 1 is the σ_{xx} (local) stress component in the element

   i = 2 is the σ_{yy} (local) stress component in the element

   i = 3 is the σ_{xy} (local) stress component in the element

   i = 4 is the ε_{xx} (local) strain component in the element

   i = 5 is the ε_{yy} (local) strain component in the element

   i = 6 is the ε_{xy} (local) strain component in the element

   The j index ranges from 1 to the number of membrane elements in the order input.
4. The local coordinate system in which stress and strain quantities are defined coincide with the global coordinate system for 2D models. For 3D models, the local coordinate system depends on the way each element is defined. The local system is shown on the appropriate **ELEM** or **SNGL** subcommand page.

PROP (properties) Subcommand
MBRN (membrane) Input Group **MBRN-PROP**

Purpose: To define material properties for membrane elements.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **MBRN** Group: Must precede any **ELEM** or **SNGL** subcommand that references the defined material model.

PCOM SCOM input parameters

PROP *membname rho emodulus poisson sigyield thickness tensyield hourglas*

membname = (C) Name of the membrane material model.
No default.

rho = (F) Mass density of the membrane material.
No default.

emodulus = (F) Young's modulus of the membrane material.
No default.

poisson = (F) Poisson's ratio of the membrane material.
No default.

sigyield = (F) Yield stress of the membrane material.
No default.

thickness = (F) Thickness of the membrane element.
No default.

tensyield = (F) Yield stress of the membrane material in uniaxial tension. If input, yield surface is distorted to have different yield stress in tension, *tensyield*, and compression, *sigyield*. *tensyield* > 0.0
Default: yield stress in tension and compression are both equal to *sigyield*
(i.e., von Mises surface)

hourglas = (F) Hourglass mode suppression constant.
Default = .01

ELEM (elements) Subcommand
MBRN (membrane) Input Group **MBRN-ELEM**

Purpose: To define the connectivity for a group of membrane elements and specify the associated material model. Membrane elements are coincident with continuum element faces.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MBRN** Group: Must follow the **PROP** subcommand that defines the membrane material properties for the defined elements.

PCOM	SCOM	input parameters
	ELEM	*membname ibegin iend jbegin jend kbegin kend*

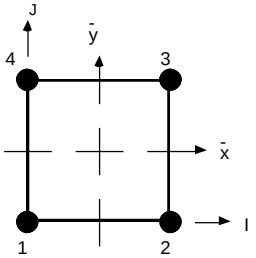
membname = (C) Name of the membrane material model.
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the group of membrane elements defined.
Default: *ibegin* = 1, *iend* = maximum I

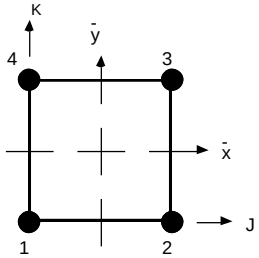
jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the group of membrane elements defined.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes that bound the group of membrane elements defined.
Default: *kbegin* = 1, *kend* = maximum K

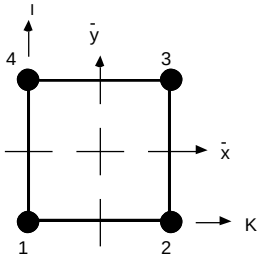
Notes:
1. *ibegin* must equal *iend, jbegin* must equal *jend,* or *kbegin* must equal *kend.*
2. The four nodes which define each membrane element are assumed to be coplanar.
3. The local coordinate system in which stress and strain quantities are defined coincide with the global coordinate system for 2D models. For 3D models, they are defined as follows:



I-J face element



J-K face element



K-I face element

SNGL (single) Subcommand
MBRN (membrane) Input Group **MBRN-SNGL**

Purpose: To define an arbitrary nodal connectivity for a single membrane element and specify the associated material model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MBRN** Group: Must follow the **PROP** subcommand that defines the membrane material properties for the defined element.

PCOM	SCOM	input parameters
	SNGL	*membname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*

membname = (C) Name of the membrane material model. No default.

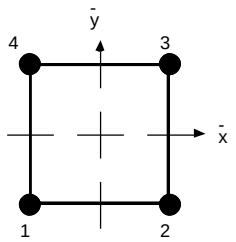
i1, j1, k1 = (I) I, J, and K indices of node 1 for the membrane element. No default.

i2, j2, k2 = (I) I, J, and K indices of node 2 for the membrane element. No default.

i3, j3, k3 = (I) I, J, and K indices of node 3 for the membrane element. No default.

i4, j4, k4 = (I) I, J, and K indices of node 4 for the membrane element. No default.

Notes:
1. The indices k1, k2, k3, and k4 are disregarded for 2D models. The proper order of inputs must be maintained, however, by providing a value for these parameters or by placing the * symbol at their positions on the input line.
2. The four nodes that define the membrane element are assumed to be coplanar.
3. The local coordinate system in which stress and strain quantities are defined coincides with the global coordinate system for 2D models. For 3D models, they are defined as follows:



Local coordinate system for **SNGL** defined membrane elements

PRNT (print) Subcommand
MBRN (membrane) Input Group **MBRN-PRNT**

Purpose: Print membrane element information.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MBRN** Group: None.

PCOM SCOM input parameters

PRNT *option*

option = (C) Print option.
Any of: PROP = print material property models

ELEM = print element connectivity and material type

ALL = print both material property and element data

Default = ALL

MEM (memory) Input Command

MEM

Purpose: To change the memory allocation available to the current job from its standard default value.

Use Is: Optional. If not input, the default memory allocation is made for the job.

Multiple Use: Not allowed.

Order Dependence:Must be first input command.

PCOM input parameters

MEM *memory_numeric memory_character option*

memory_numeric = (I) Number of millions of numeric data words to be allocated in RAM memory for this job. The number of bytes allocated is *memory_numeric* * 4 * 10^6 bytes. No default.

memory_character = (I) Number of thousands of 20 character data entries to be allocated in cpu memory for this job. The number of bytes allocated is *memory_character* * 20 * 10^3 bytes. No default.

option = (C) In general, users may ignore this input.
Set to NOPR (no print) to suppress the memory allocation message that is sent to the terminal screen when memory is allocated at the start of a run.
Set to DEBG (debug) to display some debug diagnostics.

Notes:
1. Default values for *memory_numeric* and *memory_character* can be established within the PZFlex.defaults file.
2. Restart jobs automatically restart with the memory allocation of the initial execution, providing that a MEM command was not input and the REST (restart) command is the first command of the restart run.
3. Customers with the Microsoft Windows version of PZFlex do not have to use the MEM command to allocate memory at the start of a run. The PZFlex allocates numeric memory as needed, up to the maximum physical RAM available. Character data must still be allocated with the MEM command if an amount greater than the default is required.

MGR (manager) Input Command

MGR

Purpose: To check and print out the current Data Manager tables. This command is of use only to users who are familiar with the **SOFSHEL** software architecture used in **PZFlex**.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM _____ input parameters _____

MGR *option (option2)*

option = (C) Option. Any of: GEO, CHEK, ALL or specific Data Manager array name.
ALL - the Manager tables are printed for all data arrays.
GEO - Geometry Manager information is printed for all data arrays.
CHEK - check the entire Data Manager for invalid real numbers.

If *option* is set to a specific data array name, the Manager tables only for array that are displayed.
Default = ALL.

option2 = (C) Option 2. For *option*-CHEK, set *option2* = ALL to print all invalid real numbers. Default is to print only a subset..

MODL (Model) Input Command

Purpose: To define a set of specialized behavior models which alter the stress-strain relations for specified continuum elements of the grid. These models provide the ability to simulate slip, debonding-recontact, and rebar reinforcement within continuum elements which would not be properly accounted for using the material constitutive relations assigned to the element.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow the **GRID** command and precede the **PRCS** command.

PCOM	SCOM	input parameters
MODL		
	SLIP	*modelname cohesion1 tangent1 cohesion2 tangent2*
	REBR	*modelname emodulus sigyield ratio1 ratio2 ratio3 viscous vshape*
	DBND	*modelname slipname tcutoff*
	FLUD	*modelname phi0 bulks n e tenf umax sat*
	PCEL	*modelname presname xscale yscale*
	VCTR	*vectorname vx1 vy1 vz1 vx2 vy2 vz2*
	ELEM	*modelname vectorname ibegin iend jbegin jend kbegin kend matnam*
	PRNT	*option iaxis ibegin iend jbegin jend kbegin kend*
	END	

Notes: 1. The name (*modelname*) assigned to each model must be distinct from all other names assigned to the other models input.

SLIP Subcommand
MODL (Model) Input Group **MODL-SLIP**

Purpose: To define a slip model. A slip model places limits on the shear strength of an element in addition to the limit implicitly applied through the constitutive relations of the elements material. This option provides a way to simulate slip occurring at the interface between soil and structure. The appropriate slip model and local orientation are associated with continuum elements by using the **ELEM** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: Must precede any **ELEM** subcommand which references the slip model.

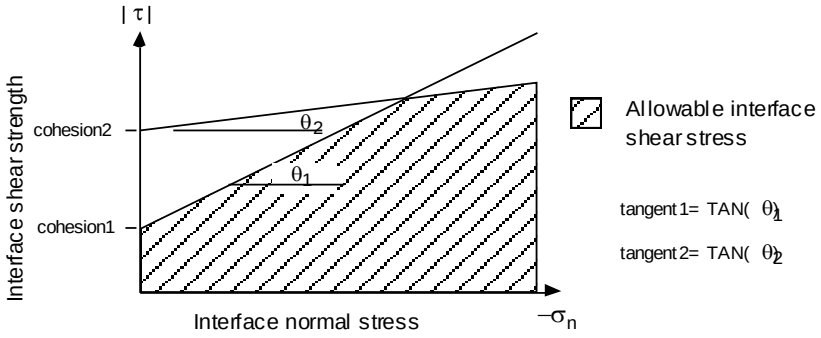
PCOM SCOM input parameters

SLIP *modelname cohesion1 tangent1 cohesion2 tangent2*

modelname = (C) Model name which uniquely identifies this model. No default.

cohesion1, tangent1 = (F) The cohesion and tangent values which define the first bounding surface to the interface shear strength as a function of interface normal stress. No default.

cohesion2, tangent2 = (F) The cohesion and tangent values which define the second bounding surface to the interface shear strength as a function of interface normal stress.
Default: *cohesion2*=*cohesion1*, *tangent2*=*tangent1*.



Notes: 1. The interface normal stress is defined to be the normal stress acting in the direction of vector 1 (defined by the associated **VCTR** subcommand). The shear stress acting tangent to the vector 1 direction is constrained to be within the allowable shear stress envelope.

REBR (Rebar) Subcommand
MODL (Model) Input Group **MODL-REBR**

Purpose: To define a rebar model. A rebar model represents the anisotropic stiffness properties due to rebar reinforcement of a continuum element in up to three orthogonal directions. The orientation of the rebar within an element is arbitrary and is defined by an associated **VCTR** subcommand. The appropriate rebar model and local orientation are associated with continuum elements by using the **ELEM** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: Must precede any **ELEM** subcommand which references the rebar model.

PCOM SCOM input parameters

REBR *modelname emodulus sigyield ratio1 ratio2 ratio3 npairs &*
edot(1) sig(1) edot(2) sig(2) ...edot(npairs) sig(npairs)

modelname = (C) Model name which uniquely identifies this model. No default.

emodulus = (F) Young's modulus of the rebar. No default.

sigyield = (F) Uniaxial yield stress of the rebar. No default.

ratio1, ratio2, ratio3 = (F) Ratio of cross-sectional area of reinforcement to total cross-sectional area of the element in directions 1, 2 and 3 defined by an associated **VCTR** subcommand. Default = 0.0

npairs= (F) Number of points ($\leq$ 10) defining rate dependent behavior in uniaxial tension. Set to 0.0 for rate independent material. Default = 0.0

edot, sig= (F) Pairs of plastic strain rate vs axial stress in uniax tension

(description of **REBR** subcommand continues on next page)

(continued)

Notes:
1. The rebar is assumed to pass through the centroid of a continuum element and its strain state is defined by the element centered strain of the continuum element.
2. The rebar is assumed to be unaffected by its lateral confinement.
3. The mass of the rebar is neglected when computing lumped nodal masses for the continuum nodes (i.e. the nodal masses assume a uniform continuum material).
4. Computation of internal resisting forces due to continuum stresses do not account for the reduced cross sectional areas of the continuum by the associated rebar areas.
5. The increased stiffness of the continuum element due to the inclusion of rebar is accounted for when determining the allowable time step size for the element.
6. The axial rebar stresses in the 1, 2 and 3 directions (defined by an associated **VCTR** subcommand) are stored in the RBR1(i,j,k), RBR2(i,j,k) and RBR3(i,j,k) data arrays, respectively. The ijk indices of these arrays correspond to elemental indices.
7. Rebar input all goes on one line. Use the continuation character (&) as necessary.

DBND (Debond) Subcommand
MODL (Model) Input Group **MODL-DBND**

Purpose: To define a debond model. A debond model places a limit on the tensile strength of a continuum element in a prescribed direction. This model provides a way of simulating debonding and recontact at a soil-structure interface. The appropriate debond model and local orientation are associated with continuum elements by using the **ELEM** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: Must precede any **ELEM** subcommand which references the debond model.

PCOM SCOM input parameters

DBND *modelname slipname tcutoff*

modelname = (C) Model name which uniquely identifies this model.
No default.

slipname = (C) A slip model name which is to control slip behavior at times when an element is not debonded. If default, no slip model is associated with this debond model.

tcutoff = (F) The tensile normal stress which defines the tensile strength of the element in the direction of vector 1 (defined by an associated **VCTR** subcommand).
Default = 0.0

Notes: 1. The computation of the gap strain and resulting element stress state assumes a uniaxial gap strain occurs in the direction of vector 1 and that the element is elastic. It is recommended that the debond element always be assigned elastic material properties, including anisotropic elastic. It must not be used with the CONC material type.

2. The gap strains which form as a result of debonding are stored in the DGAP(i,j,k) data array where the ijk indices of this array corresponds to elemental indices.

3. The DGAP data array is initialized to 0.0 during the **PRCS** command (i.e. all debond elements are initially unseparated. If it is desired to begin an analysis with one or more elements already debonded, the user can use the **SET** command to initialize the gap strain of the appropriate elements to the desired initial gap strain values.

Example: For a 2-D model, the following command placed after the **PRCS** command will set the initial gap strain of element i=5, j=8 to a value of 1 percent strain.

SET DGAP .01 5 5 8 8

FLUD (Fluid) Subcommand
MODL (Model) Input Group **MODL-FLUD**

Purpose: To define a fluid model. A fluid model adds pore fluid pressure response to an element, thus permitting an effective stress modeling approach. The appropriate fluid model is associated with continuum elements by using the **ELEM** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: Must precede any **ELEM** subcommand which references the fluid model.

PCOM SCOM input parameters

FLUD *modelname phi0 bulks n e tenf umax sat*

modelname = (C) Model name which uniquely identifies this model. No default.

phi0 = (F) Initial porosity ie, volume of pore space/ total volume (0<phi0<1). No default.

bulks= (F) Bulk modulus of the solid grains. No default.

n,e = (F) Fluid equation of state parameters. See note 5. No default

tenf = (F) Bulk modulus of fluid in tension. No default.

umax= (F) Upper bound on expected fluid pressure. No Default.

sat = (F) Saturation (ie, volume of fluid / volume of pore space). No Default.

(description of **FLUD** subcommand continues on next page)

FLUD (Fluid) Subcommand
MODL (Model) Input Group

MODL-FLUD
(continued)

Notes:
1. The fluid model adds pore fluid pressure to the continuum element element stresses, thus permitting an effective stress analysis using any of the continuum material models.
2. The effective strains of the solid skeleton are assumed to equal the total imposed strain (additional skeleton strains due to pore pressure are neglected). The fluid equation of state and porosity are iterated to satisfy equilibrium with the solid grains. The ESCP material model includes the skeleton strains due to pore pressure and is more CPU intensive.
3. The fluid mass is neglected when computing lumped nodal masses for the continuum nodes . The fluid mass should be accounted for by increasing the material density on the MATR PROP subcommand.
4. The increased stiffness of the continuum element due to the inclusion of fluid is accounted for when determining the allowable time step size for the element. The fluid equatiion of state is linearized for pressures exceeding *umax* to guarantee stability.
5. The fluid equation of state is: $u = e\,(\rho/\rho_{sat})^n - f$. ρ_{sat} is the fluid density at saturation. f is internally computed.
6. The fluid pore pressure and the element volume ratio are stored in the PFLU(i,j,k), and VFLU(i,j,k) data arrays, respectively. The ijk indices of these arrays correspond to elemental indices.
7. FLUD input all goes on one line. Use the continuation character (&) as necessary.

PCEL (Pressure cell) Subcommand
MODL (Model) Input Group

MODL-PCEL

Purpose: To override the stress-strain relations and prescribe the time variation of the σ_{xx} and σ_{yy} stress components within specified elements of the mesh. In general, the **PLOD** command is the recommended way of applying pressure loads within the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: Must precede any **ELEM** subcommand which references the pressure cell model.

PCOM SCOM input parameters

PCEL *modelname presname xscale yscale*

modelname = (C) Model name which uniquely identifies this model.
No default.

presname = (C) Name of pressure time history which defines the element's stress state. Set to FUNC if the **FUNC** command is used to define the prescribed time history. Set to name of time history if the **DATA HIST** command is used to define a digitized time history.
No default.

xscale = (F) Scale factor to use for the σ_{xx} stress component.
$\sigma_{xx}(t) = -xscale * presname(t)$
Default = 0.0

yscale = (F) Scale factor to use for the σ_{yy} stress component.
$\sigma_{yy}(t) = -yscale * presname(t)$
Default = 0.0

Notes: 1. Pressure cell models should only be assigned to elastic or void continuum elements.

2. The σ_{zz} stress component for a pressure cell element is always set to 0.0.

VCTR (Vector) Subcommand
MODL (Model) Input Group **MODL-VCTR**

Purpose: To define a local coordinate system to be assigned to a continuum element. The local system is defined by specifying a set of vectors which are used to generate three sets of orthogonal direction cosines representing local directions 1, 2 and 3. This local orientation along with various slip, rebar and debond models are associated with continuum elements using the **ELEM** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence Within **MODL** Group: Must precede any **ELEM** subcommand which references this local coordinate system.

PCOM SCOM input parameters

VCTR *vectorname vx1 vy1 vz1 vx2 vy2 vz2*

vectorname = (C) Model name which uniquely identifies the local reference system defined by the input vectors. No default.

vx1, vy1,vz1 = (F) The global x, y and z components of vector 1 (v1) which defines direction 1. No default.

vx2, vy2,vz2 = (F) The global x, y and z components of vector 2 (v2). Only input for 3-D models. No default.

Notes: 1. For 2-D models: only *vx1* and *vy1* are input by the user. Directions 1, 2 and 3 of the local system are defined as:

Direction 1 (d1) is defined as d1 = v1 (vz1 = 0.0 is assumed)
Direction 3 (d3) is defined as the global z axis
Direction 2 (d2) is defined as d2 = d3 x d1

2. For 3-D models: *vx1, vy1, vz1, vx2, vy2* and *vz2* are input by the user. Directions 1, 2 and 3 of the local system are defined as:

Direction 1 (d1) is defined as d1 =v1
Direction 3 (d3) is defined as d3 = d1 x v2
Direction 2 (d2) is defined as d2 = d3 x d1

ELEM (Element) Subcommand
MODL (Model) Input Group **MODL-ELEM**

Purpose: To assign a model and local reference system to a continuum element.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: Must follow any model or **VCTR** subcommands which it references.

PCOM SCOM input parameters

ELEM *modelname vectorname ibegin iend jbegin jend kbegin kend (matnam)*

modelname = (C) Name of model to assign to elements.
Default is the last *modelname* input on a **SLIP**, **REBR**, **DBND**, **PCEL** or **ELEM** subcommand.

vectorname = (C) Name of vector (local reference) system to assign to elements.
Default is last *vectorname* input on either a **VCTR** or **ELEM** subcommand.

ibegin, iend = (I) The beginning and ending I-indices of nodes bounding the region of elements to be assigned the model properties.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes bounding the region of elements to be assigned the model properties.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes bounding the region of elements to be assigned the model properties.
Default: *kbegin* = 1, *kend* = maximum K

matnam = *(C)* If input, only assign model to elements with material name *matnam* within the specified ijk region.
Default = assign to all materials.

PRNT (Print) Subcommand
MODL (Model) Input Group **MODL-PRNT**

Purpose: To provide the user with a way of displaying all information input within the **MODL** input group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MODL** Group: None.

PCOM SCOM input parameters

PRNT *option iaxis ibegin iend jbegin jend kbegin kend*

option = (C) Print option. No default.
Any of:

VCTR = print a map of the vector assignments
MODL = print a map of the model assignments
COS = print the direction cosines for each vector system
SLIP = print the slip model definitions
REBR = print the rebar model definitions
DBND = print the debond model definitions
PCEL = print the pressure cell model definitions

iaxis = (C) If using the VCTR or MODL print option, the printed map will be displayed for each 2-D plane of the requested region of the grid preceding in the *iaxis* direction. Any of I, J or K. Default = K.

ibegin, iend = (I) If using the VCTR or MODL print option, the beginning and ending I-indices of nodes bounding the region to be displayed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) If using the VCTR or MODL print option, the beginning and ending J-indices of nodes bounding the region to be displayed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) If using the VCTR or MODL print option, the beginning and ending K-indices of nodes bounding the region to be displayed.
Default: *kbegin* = 1, *kend* = maximum K

MODS (modifications) Input Command

Purpose: To modify the problem definition for a 2D or 3D large deformation problem. This command group is specialized and has many restrictions upon its use, see notes at the bottom of each subcommand for a description of command restrictions.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must be placed after the **EXEC** commands that execute the model up to the time at which the model modifications are to occur. An exception to this is the **GRUP**, **GEND,** and **TINC** commands which define modifications to be performed at prescribed intervals. These may be placed before or after the **PRCS** command.

PCOM SCOM input parameters

MODS

OPTN	*elmopt erodopt erodval plodopt erodmat*
EROD	*ibegin iend jbegin jend kbegin kend*
PLOD	*option ibegin iend jbegin jend kbegin kend*
PRNT	*print_option*
GRUP	*grpname*
GEND	
TINC	*grpname inctime*
TIMCHK	*option*
STOP_NODES	*option disprange*
END	

Note: Statistics are accumulated in the ERODED array as listed below.

Index	Contents
1	Total number of eroded elements
2	Eroded continuum elements
3	Eroded shell elements
4	Eroded beam elements
5	Eroded bar elements
6	Eroded Membrane elements
7	Total bad (zero-timestep) elements
8	Bad continuums
9	Bad shells
10	Bad beams
11	Bad bars
12	Bad membranes

OPTN (options) Subcommand
MODS (modifications) Input Group **MODS-OPTN**

Purpose: To define the erosion criteria for continuum or structural elements and the procedure for handling pressure loading (applied using **PLOD** commands) once an element has cracked and lost all structural integrity.

Use Is: Optional. An **OPTN** subcommand must be entered, however, before any **EROD** subcommands.

Multiple Use: Allowed. **OPTN** subcommands do not accumulate. Each **EROD** subcommand applies only to those options set by the last **OPTN** subcommand.

Order Dependence within **MODS** Group: Must precede all **EROD** subcommands that are to use the specified options.

PCOM	SCOM	input parameters
	OPTN	*elmopt erodopt erodval plodopt erodmat*

elmopt= (C) Element type(s) to be eroded. Any of: CONT, ALL, STRC, SHEL, BEAM, BAR, MBRN, JOINT.
Default = ALL.
CONT = Continuum elements
ALL = All element types
STRC = All structural elements (shell, bar, beam, membrane)
SHEL = Shell elements
BEAM = Beam (**BM3D**) elements
BAR = Bar elements
MBRN = Membrane (membrane) elements
JOINT = Line element joint springs

erodopt= (C) Quantity governing erosion. Any of: PSTN, PSDM, EVDM, GAPV, EPLS, ERGF, DACT, TSTP, ARAY (see table below).
Default = DACT.
PSTN = Principal tensile strain
PSDM = Principal tensile strain greater than threshold AND damage greater than 0.98 (SFT1, PLIH materials only)
EVDM = Volumetric strain greater than threshold AND damage greater than 0.98 (SFT1 materials only)

GAPV = Volumetric gap strain (SFT1 materials only)
EVLM = SHEL in-plane volume strain
ELDM = SHEL in-plane volume strain AND damage greater than 0.98 (PLIH, SFT1 materials only)
EPLS = Equivalent plastic strain
EVPL = Tensile plastic volumetric strain
ERGF = Fraction of available fracture energy (0<ERGF<1).
DACT = Deactivate immediately. No quantity governs erosion.
SHER = Equivalent shear strain
TSTP = This option refers to model timestep constraints controlled by the *rationmin* parameter on the **TIME** command. An element is eroded if its timestep falls below the timestep = *erodval* * Tmin (the minimum allowable time step for model). Tmin = *ratiomin* * *timestepn,* where *timestepn* is the controlling timestep when the TIME command was entered. For example: assume: *timestepn* = 0.02, ratiomin = 0.1, erodval = 3.0, then an element will erode if its timestep falls below .006.
SLIP = relative shear slip
MASS = erode springs with zero mass at an end node

erodval= (F) Value of erosion quantity that triggers erosion. Default = (0.03 for PSTN, EVPL, GAPV, EVDM or PSDM, 0.10 for EPLS, 0.70 for ERGF, 1.125 for TSTP, 0.03 for EVLM, 0.03 for ELDM)

plodopt = (C) Option defining the procedure for handling **PLOD** commands which apply pressure to continuum elements that erode. When *elmopt* is any structural element type, the *pldopt*=DACT is always assumed. See Note 4.
Any of: MOVE, DACT, NCHK. Default = MOVE.

MOVE = the **PLOD** specification is moved through the eroded element and is applied to the next active continuum element which occurs within the window specified on the **EROD** subcommand. Nodal forces are moved accordingly. Structural elements are ignored.

DACT = the **PLOD** specification is deactivated. Any forces which had been present on the nodes of the element due to the applied pressure are removed.

NCHK = This option applies only to pressure loads applied to shell elements. If this option is set, then the erosion of a shell element to which pressure loads have been applied will cause the deactivation of all **PLOD** specifications that are applied over the loaded shell surface. No check for any other remaining coincident shells or continuum sufaces is performed. Any forces which had been present on the nodes of the element due to the applied pressure are removed. This option is primarily provided to support GCON (general connectivity) shell models.

erodmat = (C) If entered, only elements of material name *erodmat* will be eroded. Default: all material types will be eroded. For shells, enter the layer definition name, not the material.

Notes:
1. A continuum element that erodes or is deactivated becomes a void element and its stress is set to 0.0. The element's nodal mass, however, is left unchanged. Structural element stresses are set to zero, and a flag is set to prevent further processing.
2. This command supercedes the obsolete **CRAK** subcommand. The **CRAK** command is retained only for upward compatiblity.
3. Beams and shells are eroded when the average of all layers exceedes the erosion criterion.

If a shell/membrane element with an applied pressure loading is eroded and there is no coincident shell/membrane elements and no continuum element on either side of the element in IJK-space, the applied pressure loading will be deactivated. If a beam/bar structural element with an applied line pressure load is eroded, the applied pressure loading is deactivated if there is no coincident beam/bar elements.

Summary of Erosion Options

ELEMENT TYPE	CONT	SHEL	BEAM	BAR	FUSE	MEMBRANE	JOINT SPRING	BOLT SPRING
OPTN/PSTN	X	X		X		X		
OPTN/EVDM	X+	X+	X+					
OPTN/GAPV	X+							
OPTN/PSDM	X++	X++						
OPTN/ERGF	X*	X	X	X		X		
OPTN/TSTP	X	X	X	X	X	X		
OPTN/EVPL		X+	X+					
OPTN/EPLS	X**	X	X	X		X		X
OPTN/DACT	X	X	X	X	X	X	X	X
OPTN/SLIP						X	X	
OPTN/SHER	X				X			
OPTN/MASS							X	
OPTN/EVLM		X						
OPTN/ELDM		X						
PLOD/MOVE	X	X				X		
PLOD/DACT	X	X				X		
PLOD/NCHK		X				X		

\+ SFT1 materials only
++ SFT1, PLIH materials only
* SFT2, PLIH materials only
** ISOH, PLIH materials only

EROD (erode) Subcommand
MODS (modifications) Input Group **MODS-EROD**

Purpose: To erode elements within the specified (i,j,k) window(s) based on options set by the most recent **OPTN** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MODS** Group: Must follow the appropriate **OPTN** subcommand.

PCOM	SCOM	input parameters
	EROD	*(araynam) ibegin iend jbegin jend kbegin kend option*

araynam = (C) Data manage array to erode on for the ARAY option I

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the region of the model to be modified.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the region of the model to be modified.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the region of the model to be modified.
Default: *kbegin* = 1, *kend* = maximum K

option = (C) Optional input parameter. If set to: CONF, then, when eroding continuum elements, the program will allow to erode only elements that meet the erosion criteria and that are not confined on all sides by other continuum elements or quadralateral structural elements such as shells and membranes. This option is ignored when eroding any non-continuum element type. See Note 4.

Notes:
1. A continuum element that erodes or is deactivated becomes a void element and its stress is set to 0.0, however, the element's nodal mass is left unchanged. Structural element stresses are set to zero, and a flag is set to prevent further processing.
2. The procedure for handling pressure loads applied to elements that have been removed from the model is specified by the **OPTN** subcommand.
3. This command supercedes the obsolete **CONT** subcommand.
4. The CONF (confinement) option logic looks at adjacent continuum elements in IJK-space in order to determine whether an element is confined. Bond connections are not considered in this logic. An element with at least one adjacent element that is void is considered unconfined unless a structural element covers the element's free faces.

PLOD (pressure load) Subcommand
MODS (modifications) Input Group **MODS-PLOD**

Purpose: To modify the pressure loading (**PLOD**) specifications for a region of the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MODS** Group: None.

PCOM	SCOM	input parameters
	PLOD	*option ibegin iend jbegin jend kbegin kend*

option = (C) Option to use for modifying the pressure loading applied to elements within a region of the model.
Any of: DACT

DACT = Deactivate all pressure loading on elements falling within the specified ijk window.

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the region of the model to be modified.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the region of the model to be modified.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the region of the model to be modified.
Default: *kbegin* = 1, *kend* = maximum K

Notes:
1. This option can be used for any model including small deformation.
2. Any **PLOD** surface segment completely within the ijk-indices window specified will be deactivated.
3. A deactivated **PLOD** segment will have any previously applied pressure removed at the time the **MODS PLOD** command is input. No pressure is applied to this surface segment after it is deactivated.

PRNT (print) Subcommand
MODS (modifications) Input Group **MODS-PRNT**

Purpose: To control the amount of printed output produced by the **MODS** command group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MODS** Group: The printed output option selected with the **PRNT** subcommand applies to all following **MODS** commands.

PCOM SCOM input parameters

PRNT *print_option*

print_option = (C) Print option to use.
Any of: ON - full printing is on
OFF - no print out is produced
EROD - only print information about each eroded element is produced
Default = ON

GRUP (group) Subcommand
MODS (modifications) Input Group **MODS-GRUP**

Purpose: To initiate the definition of a group of modifications that are to be performed at prescribed time intervals. The **GEND** subcommand is used to terminate the group definition. **OPTN** and **EROD** subcommands constitute the group and the **TINC** subcommand defines the time interval for executing the group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MODS** Group: Must precede the **OPTN**, **EROD,** and **GEND** subcommands for this group.

PCOM	SCOM	input parameters
	GRUP	*grpname*

grpname = (C) Unique name for this group. No default.

Notes:
1. Group definitions are cumulative. To deactivate a group, set its time interval to a large number with the **TINC** subcommand.
2. Each group must begin with an **OPTN** subcommand.
3. Example:

```
MODS
     GRUP SET1
     OPTN CONT PSTN 0.01
     EROD 1 100 1 50 1 5
     EROD 150 200 1 50 1 5
     GEND
     TINC SET1 15
```

GEND (group end) Subcommand
MODS (modifications) Input Group **MODS-GEND**

Purpose: To terminate the definition of a group of modifications that are to be performed at prescribed time intervals.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MODS** Group: Must follow the **OPTN**, **EROD**, and **GEND** subcommands for this group.

PCOM	SCOM	input parameters
	GEND	

TINC (time increment) Subcommand
MODS (modifications) Input Group **MODS-TINC**

Purpose: To define or change the time interval for executing a group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MODS** Group: Must follow the **OPTN, EROD, and GEND** subcommands for the referenced group.

PCOM	SCOM	input parameters
	TINC	*grpname inctime*

grpname = (C) Group name. No default.

inctime = (I) Timestep interval, eg for *inctime* = 15, the group is evaluated every 15th timestep. Default = 1.

STOP_NODES Subcommand
MODS (modifications) Input Group **MODS-STOP_NODES**

Purpose: To deactivate nodes outside of the area of interest.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must be placed after the **PRCS** command if not within a **GRUP**.
within **MODS** Group: None.

PCOM SCOM input parameters

STOP_NODES *option disprange*

option = (C) Any of:
STOP: sets the translational and rotational nodal velocities and forces to zero.
KILL: sets the translational and rotational nodal velocities and forces to zero and zeroes the nodal mass.
No Default.

disprange = (F) Displacement range. Any nodes with displacements greater than *disprange* are deactivated. No Default.

Notes:
1. The **STOP_NODES** subcommand can be placed within a **GRUP** structure to check for nodes outside of the displacement range at specified timestep increments or can be run immediately during execution if it is entered outside of the **GRUP** structure.
2. To prevent elements containing nodes that are stopped from excessive distortion when not all of the nodes in the element are stopped at the same time, the timestep increment set in **TINC** should be small. This way all nodes in the element are stopped before large distortions occur. Alternatively, the **STOP_NODES** subcommand can be used in conjunction with element erosion set for an equivalent element-centered displacement.

TIMCHK (timestep check) Subcommand
MODS (modifications) Input Group **MODS-TIMCHK**

Purpose: To avoid unnecessary timestep recalculation.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MODS** Group: None.

PCOM SCOM input parameters

TIMCHK *option*

option = (C) Timestep check. Any of OFF or ON. Default = ON.

Note: By default, MODS recalculates all element timesteps. In severe environments, this permits erosion of badly deformed elements (via the timestep criterion) before they influence the model timestep. If erosion on timestep is not to be used, or if the loading is slow enough that several timesteps may elapse before eroding, then the timestep recalculation can be deactivated. This reduces CPU cost.

MP (multiprocessing) Input Command

MP

Purpose: To set parameters governing shared memory parallel processing on systems for which it is supported.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM SCOM input parameters

MP

OMP *maxthreads dynoption*

END

OMP Subcommand
MP (multiprocessing) Input Group **MP-OMP**

Purpose: To specify parameters for Open MP symmetric multi-processing.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **MP** Group: None.

PCOM	SCOM	input parameters
	OMP	*maxthreads dynoption*

maxthreads = (I) Maximum number of threads to use for this job. Default = maximum number of processors available on the system.

dynoption = (C) Either of STAT or DYN. DYN allows the Operating System to vary the number of threads between 1 and *maxthreads*. STAT requires *maxthreads* threads for the duration of the job. Default = DYN.

OBJT Input Command

Purpose: To define geometric objects that can be used by other commands to identify elements.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any command that references the object.

PCOM	SCOM	input parameters
OBJT		
	FLIT	*objectname axisname Ox Oy Oz A B C optn & ibegin iend jbegin jend kbegin kend*
	CIRC	*objectname Ox Oy Oz Ay Ax Az Bx By Bz thick & ibegin iend jbegin jend kbegin kend*
	END	

FLIT (fillet) Subcommand
OBJT Input Group **OBJT-FLIT**

Purpose: Defines a filet object.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **OBJT** Group: None.

PCOM	SCOM	input parameters
	FLIT	*objectname axisname Ox Oy Oz A B C optn & ibegin iend jbegin jend kbegin kend*

CIRC (circle) Subcommand
OBJT Input Group **OBJT-CIRC**

Purpose: Defines part of a circular region in 2D, cylinder in 3D.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **OBJT** Group: None.

PCOM SCOM input parameters

CIRC *objectname Ox Oy Oz Ay Ax Az Bx By Bz thick &*
ibegin iend jbegin jend kbegin kend

objectname = (C) Name of the object definition.
No default.

Ox Oy Oz = (F) X,Y, and Z coordinates of point O.

No default.

Ax Ay Az = (F) X,Y, and Z coordinates of point A.

No default.

Bx By Bz = (F) X,Y, and Z coordinates of point B.

No default.

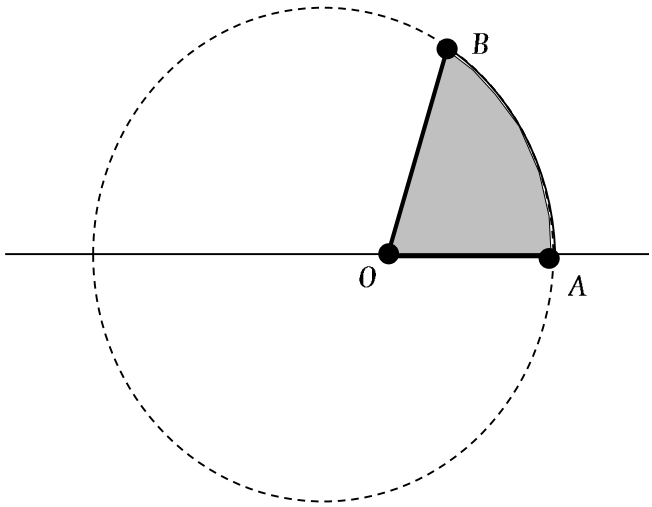
thick = (F) Thickness of the cylinder in the direction normal to the axes defined by O,A,B.

Default = 1.

ibeg, iend = (I) Beginning and ending nodal I-indices bounding the search region. Default: *ibeg* = 1, *iend* = maximum I

jbeg, jend = (I) Beginning and ending nodal J-indices bounding the search region. Default: *jbeg* =1, *jend*=maximum J

kbeg, kend = (I) Beginning and ending nodal K-indices bounding the search region. Default: *kbeg*=1, *kend*=maximum K



Notes: 1. Points O, A, and B must be distinct and noncollinear.
2. Negative values of *thick* result in the cylinder extrusion in the negative normal direction.

PIEZ (piezoelectric) Input Command

PIEZ

Purpose: To allow the option of including piezoelectric materials within the model. Currently this option is allowed only for standard partition grids containing only continuum elements.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the **GRID** command and precede the **PRCS** command when using the **WNDO**, **NODE**, **NOD2, NOD3,** and **CALC** subcommands. Should be placed after the **PRCS** command when using the **TIME** and **EXEC** subcommands. The **IMPD**, **CNVR,** and **PRNT** subcommands may occur either before or after the **PRCS** command.

PCOM	SCOM	input parameters
PIEZ		
	WNDO	*ibegin iend jbegin jend kbegin kend*
	DEFN	*electrodename*
	NODE	*ibegin iend jbegin jend kbegin kend*
	NOD2	*mat1 mat2 ibegin iend jbegin jend kbegin kend*
	NOD3	*objectname side ibegin iend jbegin jend kbegin kend*
	BC	*electrodename option histname scalehist shifthist*
	CONN	*electrodename circuitname sourceoption histname scalehist shifthist*
	SLVR	*option*
	AXIG	*arcangle factor ibegin iend jbegin jend*
	SIDE	*iside type option value(1) value(2) ... value(6)*
	CNVR	*convergence*
	CALC	*option*
	IMPD	*frequency scalecurrent scalevoltage*
	HRGL	*hourglass*
	TIME	*time*
	ESTA	*nskip*
	EXEC	*numitr*
	PRNT	*option*
	NONL	*rtol maxits*
	CUPL	*copt*
	ECON	*electrode1 electrode2 type value(s)*

WNDO (Window) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-WNDO**

Purpose: To specify the regions of the grid within which the electrostatic computation will be performed. At a minimum, a window must encompass any applied voltage conditions and all piezoelectric material for which electro-mechanical coupling is desired to be modeled.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
Within **PIEZ** Group: Must precede any **NODE** or **NOD2** subcommands.

PCOM SCOM input parameters

WNDO *ibegin iend jbegin jend kbegin kend*
or
WNDO AUTO *option*

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the electric window.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the electric window.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the electric window.
Default: *kbegin* = 1, *kend* = maximum K

option = (C) Any of DIEL or PIEZ. Default = DIEL

Notes:
1. The larger the electric window, the longer it will take to compute the electrostatic solution at each timestep.
2. More than one electric window can be declared, however, the windows must not overlap or contact each other and they will be electrically decoupled from one another.
3. Several data arrays will be created for all nodes or elements within the electric window. The i -index of these arrays ranges from 1 to the total number of nodes (or elements) summed over all windows. These are:
   POTN(i) - the nodal electric potential array
   QFRC(i) - the nodal electric charge array
   EX(i) - the x-component of each element's electric field
   EY(i) - the y-component of each element's electric field
   EZ(i) - the z-component of each element's electric field
4. AUTO defines a single electric window surrounding all DIEL or PIEZ materials.
   Note that the EX, EY and EZ arrays are only available if the user request them by using the **CALC** subcommand. The EZ array will only be available for 3-D models.

DEFN (electrode definition) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-DEFN**

Purpose: To initiate the definition of an electrode within the model. This definition is used when computing the electrostatic field. The definition of an electrode is completed by using **NODE** or **NOD2** subcommands to define the grid nodes that make up the electrode and the **CONN** subcommand to connect it to a circuit model (if any) or the **BC** subcommand to define voltage boundary conditions directly to the electrode.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **WNDO** subcommand and precede any **NODE** or **NOD2** subcommands that specify the grid nodes that make up this electrode and any **CONN** or **BC** subcommands that reference this electrode.

PCOM SCOM input parameters

DEFN *electrodename areascale*

electrodename = (C) Name of this electrode. No default.

areascale = (F) Scale factor for electrode area. Used to account for the z-dimension in 2D plane strain models and symmetry planes in 2D and 3D models. For example, if only half of an electrode's area is included within the model due to the assumption of a symmetry plane, the value of *areascale* should be increased by a factor of 2.
Default = 1.0 except for 2D axisymmetric problems which defaults to 2π since axisymmetric models evaluate a 1 radian sector of the physical problem.

Notes: 1. An example of a group of commands that define an electrode and prescribe a circuit and driving source for a 2D model with a symmetry plane intersecting the electrode is:

```
PIEZ
      WNDO  1  20  1 20      /* define the electric window
C begin definition of top electrode, the area is scaled by 2 x .012 mm
C to account for the z-dimension and the symmetry plane
DEFN TOP .024
      NODE 1  20  1  1       /* specify nodes for top electrode
      DEFN BOTM .024         /* begin definition of bottom electrode
      NODE 1  20  20  20     /* specify nodes for bottom electrode
      BC TOP GRND            /* ground top electrode
C connect bottom electrode to circuit model C1 defined by previous CIRC commands
C and define the driving function
      CONN BOTM C1 FUNC
      END
```

2. Information about each defined electrode is stored within the data array PIZE(i,j). This array is doubly subscripted. The j-index ranges from 1 to the number of electrodes specified for the model. The i-index ranges from 1 to 5 where:

   i=1 - potential voltage of the electrode at the present time
   i=2 - total charge on the electrode at the present time
   (will be scaled by *areascale* to account for total charge on electrode)
   i=3 - current , time rate of change of charge over the last time step
   i=4 - temporary work slot
   i=5 - temporary work slot

Example: Request time histories of potential voltage and charge for the third and fourth electrodes defined for the model.

```
POUT
    RATE 1
    HIST PIZE 1 2 1 3 4 1        /* four time histories will be saved
    END
```

NODE (electrode nodes) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-NODE**

Purpose: To select nodes of the model which belong to the electrode defined by the last **DEFN** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **DEFN** subcommand which initiated the definition of the electrode.

PCOM	SCOM	input parameters
	NODE	*ibegin iend jbegin jend kbegin kend*

ibegin, iend = (I) Beginning and ending I-indices of nodes that are assigned to this electrode.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that are assigned to this electrode.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that are assigned to this electrode.
Default: *kbegin* = 1, *kend* = maximum K

Notes:
1. The **NOD2** subcommand provides an alternative way of identifying electrode nodes.
2. More than one **NODE** subcommand may be used to identify an electrode's nodes.
3. A node should not be defined as an electrode node more than once.

NOD2 (Electrode nodes 2) Subcommand
PIEZ (Piezoelectric) Input Group **PIEZ-NOD2**

Purpose: To select a group of nodes of the model which fall on the interface between two specified materials and which belong to the electrode defined by the last **DEFN** subcommand . More than one **NOD2** subcommand may be used to define the nodes of an electrode. Allows a simple way of defining an electrode on a non-cartesian interface between two materials. A search window may be specified to restrict the search for interface nodes to a local region of the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **PIEZ** Group: Must follow the **WNDO** subcommand.

PCOM SCOM input parameters

NOD2 *(option) mat1 mat2 ibegin iend jbegin jend kbegin kend*

or

NOD2 *(option) mat1* **side** *iside ibegin iend jbegin jend kbegin kend*

option = (C) If ROOT is entered, all materials with names beginning with *mat1* will be compared to *mat2* or *iside*

mat1, mat2 = (C) The names of the two materials whose interface nodes will have this voltage boundary condition applied. Enter ALL for *mat2* to select all *mat1* interfaces.

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the region of the model to be searched for interface nodes. See Note 4.
Default: an I-index range 1 greater in each direction than the electric window.

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the region of the model to be searched for interface nodes. See Note 4.
Default: a J-index range 1 greater in each direction than electric window.

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the region of the model to be searched for interface nodes. See Note 4.
Default: a K-index range 1 greater in each direction than the electric window.

continued next page

PCOM SCOM input parameters

iside = (I) If keyword **side** has been entered, then nodes on *iside* of the current electric window in contact with *mat1* are added to the electrode. For this option the default index ranges coincide with the electric window

Notes:
1. The **NODE** subcommand provides and alternative way of specifying voltage boundary conditions.
2. More than one **NOD2** subcommand may be used to identify an electrode's nodes.
3. A node should not be defined as an electrode node on more than one electrode.
4. When 2 materials are entered, the ijk search window must bound the 'elements' that are on bothe sides of the interface rather than just the interface nodes themselves, i.e. the code does not look at elements that fall ouside the window defined by *ibegin, iend, jbegin*, etc.

NOD3 (electrode nodes 3) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-NOD3**

Purpose: To select a group of nodes of the model which fall on the boundary of an object defined by the **OBJT** command and which belong to the electrode defined by the last **DEFN** subcommand . More than one **NOD3** subcommand may be used to define the nodes of an electrode. Allows a simple way of defining an electrode on a non-Cartesian interface. A search window may be specified to restrict the search for boundary nodes to a local region of the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **WNDO** subcommand.

PCOM SCOM input parameters

NOD3 *objectname side ibegin iend jbegin jend kbegin kend*

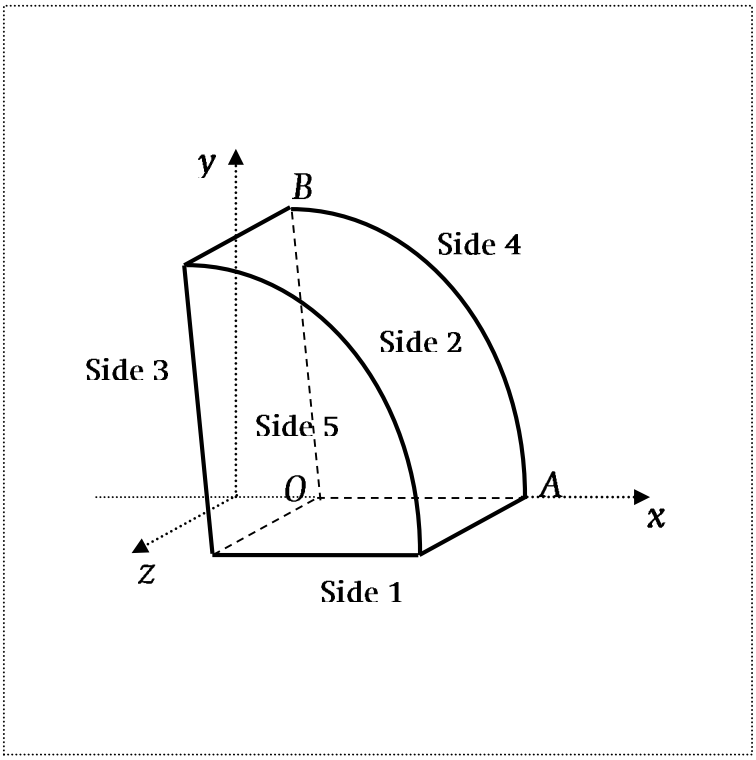
objectname = (C) Name of the object on whose surface nodes the voltage boundary condition is applied. The object must be of *objtyp=circ*.

side = (I) Side of the object on which the voltage boundary condition is applied. See diagram.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the region of the model to be searched for interface nodes. See Note 4.
Default: an I-index range 1 greater in each direction than the bounding box of the object.

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the region of the model to be searched for interface nodes. See Note 4.
Default: a J-index range 1 greater in each direction than the bounding box of the object.

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the region of the model to be searched for interface nodes. See Note 4.
Default: a K-index range 1 greater in each direction than the bounding box of the object.



O, *A* and *B* are as defined in **OBJT-CIRC**, *x, y, and z* are the local object axes.
Side 1 always lies in the local *zx*-plane along edge *OA*.
Side 2 is the curved surface.
Side 3 is formed by edge *OB* and the local z-axis.
Side 4 lies in the *xy*-plane between edges *OA* and *OB*.
Side 5 lies in the *xy*-plane a distance of the object thickness along the local *z*-axis.

Notes:
1. The **NODE** subcommand provides an alternative way of specifying voltage boundary conditions.
2. More than one **NOD3** subcommand may be used to identify an electrode's nodes.
3. A node should not be defined as an electrode node on more than one electrode.
4. The ijk search window must bound the "elements" that are on both sides of the interface rather than just the interface nodes themselves, i.e., the code does not look at elements that fall ouside the window defined by *ibegin, iend, jbegin*, etc
5. If the object surfaces do not coincide with the i,j,k node coordinates, it is possible that some corner nodes will not be assigned the voltage boundary condition when this command is used. It is best to double check that all the desired nodes have been properly assigned.

BC (electrode boundary conditions) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-BC**

Purpose: To specify voltage boundary conditions to be applied directly to an electrode. If there is a circuit connected to an electrode, the driving source is defined by the **CONN** (connect) subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **DEFN** subcommand that defines the electrode.

PCOM	SCOM	input parameters
	BC	*electrodename option histname scalehist shifthist*

electrodename = (C) Name of the electrode to which this boundary prescription applies. No default.

option = (C) Boundary condition option.
Any of: VOLT= prescribed voltage
GRND= ground (voltage set to 0.0)
OFF= turn off boundary condition
ON= turn on boundary condition

histname = (C) Name of time history which defines the prescribed constraint. Set to FUNC if the **FUNC** command is used to define the time history. Set to the name of time history if the **DATA HIST** command is used to define a digitized time history. Not needed for GRND option. No default.

scalehist = (F) Scale factor to scale the prescribed voltage or charge defined by *histname* before applying it to the electrode.
Default = 1.0

shifthist = (F) Shift value to add to the time of *histname* before applying the time history to the electrode.
Default = 0.0

Notes: 1. If *scalehist* is -1.0, the sign of the prescribed voltage is reversed. If *scalehist* is 3.0, the magnitude of the prescribed voltage is three times greater than in *histname*.
2. If *histname* defines a square pulse that begins at time 0.0, setting *shifthist* to 1.0e-6 causes the prescribed square pulse signal to begin at time = 1.0e-6.

CONN(connect) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-CONN**

Purpose: To connect a circuit and a source to an electrode.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **CIRC** command group that defines the circuit and the **DEFN** subcommand that defines the electrode being connected to the circuit.

PCOM SCOM input parameters

CONN *electrodename circuitname sourcetype histname scalehist shifthist*

electrodename = (C) Electrode name. No default.

circuitname = (C) Circuit name. No default.

sourcetype = (C) Source type. No default.
Any of: VOLT = prescribed voltage
CHRG = prescribed charge
CRNT = prescribed current
GRND = ground
OPEN = open circuit

histname = (C) Name of time history that defines applied voltage or charge at the source. Not input for GRND or OPEN options. Set to FUNC if the **FUNC** command is used to define the time history. Set to the time history name if the **DATA HIST** command is used to define a digitized time history. No default.

scalehist = (F) Scale factor to scale the prescribed voltage or charge defined by *histname* before applying it to the circuit. Not input for GRND or OPEN options. Default = 1.0.

shifthist = (F) Shift value to add to the time of *histname* before applying it to the circuit. Not input for GRND or OPEN options. Default = 0.0.

Notes:
1. Source types may be changed during the course of a calculation (e.g., a prescribed voltage may be switched to an open circuit). This discontinuous action can introduce unwanted noise into the calculation if not carefully timed.
2. Circuit element definitions may also be changed during the course of a calculation (e.g., for send or receive modes). This is done by defining a second circuit with the same number of elements as the original circuit, then issuing a **PIEZ CONN** command to connect the electrode to the second circuit at the desired time. Again, this discontinuous action can introduce unwanted noise unless the circuit is nearly quiet.
3. The charge and voltage throughout the circuits are available in the CRTQ and CRTV data arrays, respectively. These arrays are dimensioned 1 to the number of circuit nodes. There are two nodes for each circuit element. These are ordered so the node closest to the transducer appears first. A table in the flxprt file following the **PIEZ** command group lists the circuit elements and the corresponding CRTQ, CRTV indices. Time histories of these arrays may be requested with the **POUT** command group. Note that the voltages on connected nodes are identical, but charges on connected nodes sum to zero.

SLVR(solver) Subcommand
PIEZ Input Group **PIEZ-SLVR**

Purpose: To define the time integration scheme for electrostatic solutions.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM SCOM input parameters

SLVR *slvrtype (args)*

Any of:

DRCT
DISK
SPLU * *droptol*
CGIC
CGDS * *(onedee)*
DCGD * *predopt*
CGTRI * *predir predopt*
PARD

slvrtype (C) Solver type. Any of: DRCT, DISK, CGDS, CGIC, SPLU, DCGD, CGTRI or PARD. Default = DRCT.

droptol (F) Drop tolerance for SPLU solver (see note 3 below). Default = 1.e-12

onedee (C) 1D electric field approximation. Enter *i, j or k* for a 1D electric field in the i, j, or k direction. Default = *: no approximation. See note 3.

predopt (C) Prediction option. Enter CONT to activate. This option starts each CG solve with an estimate of the solution based on extrapolating previous results. Requires a little more storage, but can yield significant reductin in iterations. Default = not active.

predir *(C)* Tridiagonal preconditioning direction, i,j,or k.

Notes: 1. The DRCT option implements an in-core solver. This is the preferred solver if the problem is small enough to fit in available RAM. The DISK option implements an out-of-core direct solver. This is somewhat slower than the in-core solver, but is limited only by available disk, not RAM. CGDS is an iterative method using conjugate gradients with diagonal scaling. It is the most efficient method in terms of RAM requirements, and does not require matrix factorization. CGIC implements conjugate gradients with an incomplete cholesky preconditioner. CGIC should be considered experimental for now because the preconditioner is recomputed each timestep. SPLU is a sparse direct solver. DCGD is the same as CGDS but uses a diagonal storage scheme. It requires more storage than CGDS but is faster and parallelization nmore efficient. Both CGDS and DCGD run in parallel on OMP systems. CGTRI implements tridiagonal preconditioning in the chosen direction.

2. The SPLU solver tries to preserve sparsity during LU factorization. First the matrix is normalized to 1.0. During factorization, entries that become smaller than the drop tolerance are removed. Larger values reduce storage requirements but reduce accuracy.
3. This option assumes that i is aligned with +x, j with +y, and k with +z. Infinite element boundary conditions are not implemented. Electric field is approximate, but can produce a dramatic reduction in computational requirements.
4. DRCT is typically fastest for small problems, DCGD for large ones.
5. PARD implements the PARDISO shared memory parallel sparse direct solver from the Intel MKL library. It is typically much faster than DRCT.

AXIG (axisymmetric circumferential gradient) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-AXIG**
(2D axisymmetric models only)

Purpose: This option is specific to axisymmetric problems with electrodes spaced uniformly in the circumferential direction and piezoelectric material poled circumferentially. The command allows these types of models to be simulated using the 2D axisymmetric modeling option as long as the mechanical response of the system is actually axisymmetric in nature. Without this command, the circumferential component of the electric field is assumed 0.0. When the command is used, the circumferential component of the electric field is computed based on the assumption that the voltage varies from the driven electrode value (defined in the 2D axisymmetric slice of the model) to a grounded electrode (voltage = 0.0) over a user-specified arc angle.

Use Is: Optional.

Multiple Use: Allowed. There can be more than one grid window defined in which circumferential electrodes produce circumferential gradients in the voltage potential. These gradients may differ from window to window.

Order Dependence
within **PIEZ** Group: None.

PCOM	SCOM	input parameters
	AXIG	*arcangle factor ibegin iend jbegin jend*

arcangle = (F) Arc angle (in degrees) that defines the angular spacing between circumferentially placed electrodes.
No default.

factor = (F) Coupling factor used to "tune" the response of the model to match actual 3D model results. A value of 1.0 implies that the assumed z-component of the electric field matches the physical problem. Using a value of .8 indicates that the effective magnitude of the z-component of the electric field is 80% of the assumed approximation. See Note 1 below.
Default=1.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the grid region in which the defined circumferential gradient information applies.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the grid region in which the defined circumferential gradient information applies.
Default: *jbegin* = 1, *jend* = maximum J

Notes: 1. When representing a piezoelectric ring with electrodes spaced circumferentially and which cut entirely through the ring material, a coupling factor of 1.0 has been found to be a good value unless experimental data shows otherwise. When the electrodes are simply wrapped around the exterior of the piezoelectric ring, instead of cutting through it, a value of .85 is a good choice unless data show otherwise.

2. The piezoelectric material should be poled in the z-direction to represent a material poled circumferentially.

SIDE Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-SIDE**

Purpose: To assign boundary conditions to an entire side of an electric window.

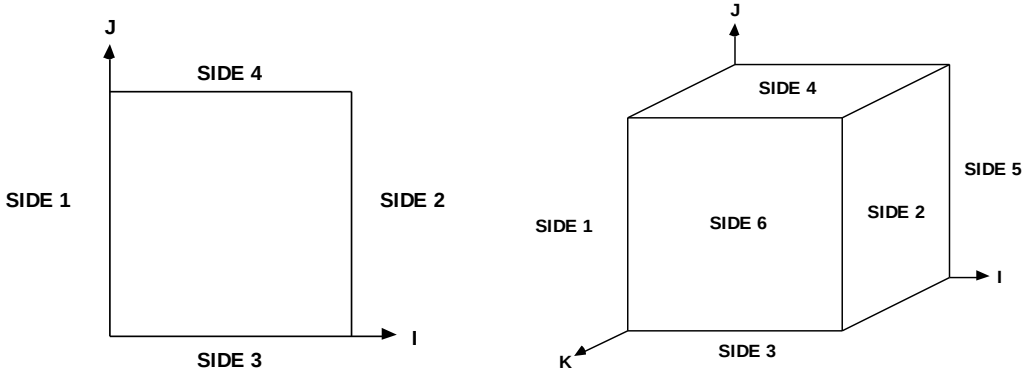
Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Must follow the **WNDO** subcommand defining the electric window to which the boundary conditions apply. Must precede other **WNDO** subcommands.

PCOM	SCOM	input parameters
	SIDE	*iside type option value(1) value(2) ... value(6)*

iside = (I) Side number specifying on which side to apply the specified boundary condition. Must be from 1 to 6 as shown.



2Dimensional models 3Dimensional models

Side number relationship to grid indices orientation

type = (C) Type of boundary conditions to be applied on side *iside*. No default.
Any of: INFN = infinite element

option= (C) Option for the type of boundary condition specified. Valid choices for *option* depend on the choice of *type*.

Boundary type	options
INFN	RADL — element diverges radially NRML — element projects normally

value(n)= (F) For INFN boundaries, *value(1), value(2), value(3)* are the (x,y,z) coordinates of the reference point (Pole). Default = center of electric window.

Notes:
1. The default boundary conditions on each electric window are that the normal component of the electric field is zero.
2. The INFN boundary condition uses "infinte elements" to terminate the grid. These approximate an infinite domain by enforcing a decay of the potential to zero at infinity. For option = RADL, the decay is radial from the reference point (e.g., for a small or point source). For option = NRML, the decay is normal to the boundary (e.g., for a large source).
3. Specified potential or equi-potential conditions can be specified via the DEFN subcommand.
4. The INFN conditions should not be used on the axis of an axisymmetric model.

CNVR (Convergence) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-CNVR**

Purpose: To allow the user to modify the convergence criteria used in defining convergence of the iterative conjugate gradient solver used for the electrostatic solution.

Use Is: Optional. If not input, the default convergence ratio is 1.e-6 and the maximum number of iterations is 1000.

Multiple Use: Allowed.

Order Dependence
within **PIEZ** Group: None.

PCOM	SCOM	input parameters
	CNVR	*convergence maxitr*

convergence = (F) Convergence criteria.
Default=1.e-6

maxitr = (I) Maximum number of iterations.
Default=1000.

Note: Once the ratio of the present residual error over the peak residual error has dropped below *convergence*, the solution is considered to have converged. If convergence is not achieved within *maxitr* iterations, a warning is printed, the current solution is accepted, and the computation proceeds to the next time step.

CALC (calculate) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-CALC**

Purpose: To request that the electric field be computed within the electric window.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **PIEZ** Group: None.

PCOM	SCOM	input parameters
	CALC	*option*

option = (C) Set to ELEC to request that the electric field be computed throughout the computation. DISP calculates the electric displacement as well as the electric field.
No default.

Notes: 1. The electric field for each element within the electric window is computed and stored in the EX(i,j,k), EY(i,j,k), and EZ(i,j,k) data arrays, where the i-, j-, and k-indices are the local elemental indices within the electric window. The EZ array is available only for 3D models.
2. Likewise, the electric displacement is stored in arrays DXPZ, DYPZ, and DZPZ. Again, DZPZ is computed only for 3D analyses.

IMPD (impedance) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-IMPD**

Purpose: To request that the steady-state impedance for each voltage boundary condition prescribed be computed. This option requires that the model be driven at a constant frequency of excitation until steady state is achieved.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: None.

PCOM	SCOM	input parameters
	IMPD	*frequency scalecurrent scalevoltage*

frequency = (F) Cycles per unit time.
No default.

scalecurrent = (F) Scale factor to scale the current by before computing impedance.
Default = 1.0

scalevoltage = (F) Scale factor to scale the voltage by before computing impedance.
Default = 1.0

Note: The scale factors *scalecurrent* and *scalevoltage* can be used to adjust impedance results. For example, when a 3D experiment is modeled with a 2D model, the 2D model assumes a unit thickness in the out of plane direction for each electrode. If *scalecurrent* is set equal to the proper out of plane dimension of the electrode, the computed results can be directly compared with actual experimental measurements.

HRGL (hourglass) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-HRGL**

Purpose: To specify that hourglass control for the electrostatic solution be activated and to set the hourglass control value.

Use Is: Optional. If not input, no hourglass suppression is used if the iterative solver is chosen but a .01 hourglass suppression constant is used if the direct solver is chosen.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: None.

PCOM	SCOM	input parameters
	HRGL	*hourglass*

hourglass = (F) Hourglass control constant. Value should be kept much less than 1.0. Typically .01 is a good choice.
Default = .01

TIME Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-TIME**

Purpose: This subcommand is used in conjunction with the **EXEC** subcommand to allow the computation of a static electric field without a dynamic analysis. It sets the time at which the prescribed voltage should be evaluated using the time histories specified for the voltage boundary conditions.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Most precede the **EXEC** subcommand. The model must already have been processed with the **PRCS** command

PCOM	SCOM	input parameters
	TIME	*time*

time = (F) Time at which to evaluate the voltage to apply
Default = 0.0

Note: The proper procedure for using the **TIME** and **EXEC** subcommands is as follows:

(a) Define the complete model, including all prescribed voltage boundary conditions, and process it using the **PRCS** command.

(b) Reenter the **PIEZ** command group and set the time at which to evaluate the voltage boundaries for the applied potential desired.

(c) Use the **EXEC** subcommand to execute a specified number of iterations. The iteration stops when convergence is achieved.

ESTA (electrostatic)Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-ESTA**

Purpose: This subcommand activates electrostatic force computations (Coulomb's law).

Use Is: Optional.

Multiple Use: Not Allowed.

Order Dependence
within **PIEZ** Group: None

PCOM	SCOM	input parameters
	ESTA	*nskip*

nskip= (I) Number of timesteps between capacitance matrix recomputation.
Default = 1, every timestep.

Notes: 1. ESTA is implemented only for large deformation analyses.
2. The capacitance matrix is reassembled each *nskip* timestep to account for geometry changes. Values larger than 1 are an approximation that can improve performance considerably when a direct solver is used. *Nskip* should be set to 1 for cMUTs nearing contact to accurately capture the nonlinearity.

EXEC (execute) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-EXEC**

Purpose: This subcommand is used in conjunction with the **TIME** subcommand to allow the computation of a static electric field without a dynamic analysis. It defines the number of iterations to execute when trying to converge to the static solution. If the solution does not converge, additional **EXEC** subcommands may be input.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: Most follow **TIME** subcommand. The model must already have been processed with the **PRCS** command.

PCOM	SCOM	input parameters
	EXEC	*numitr*

numitr = (I) Number of iterations to execute.
Default = 1

Note: See the notes for the **TIME** subcommand, which describe the proper procedure for using the **TIME** and **EXEC** subcommands.

PRNT (print) Subcommand
PIEZ (piezoelectric) Input Group **PIEZ-PRNT**

Purpose: To turn on or off the print of the number of iterations required to achieve convergence at each time step of a dynamic calculation.

Use Is: Optional. If not input, no convergence information is printed.

Multiple Use: Allowed.

Order Dependence within **PIEZ** Group: None.

PCOM SCOM input parameters

PRNT *option*

option = (C) ON, OFF, or ALL.
ON = print convergence for each time step
OFF = don't print anything about convergence
ALL = print convergence statistics for each iteration
Default = ON

NONL(nonlinear) Subcommand
PIEZ Input Group **PIEZ-NONL**

Purpose: To define parameters governing nonlinear electric field solutions.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM	SCOM	input parameters
	NONL	*rtol maxits*

rtol (F) Relative tolerance for solution of nonlinear equations; see note 1. Default = $1x10^{-6}$.

maxits (I) Maximum number of nonlinear iterations per timestep. Default = 10

Note: A residual load iteration with step-size control is used to solve the nonlinear electric field relatons, e.g., when the model contains electrostrictive materials . Iterations at each timestep continue until either the relative error is less than *rtol* or *maxits* has been exceeded.

CUPL(Couple) Subcommand
PIEZ Input Group **PIEZ-CUPL**

Purpose: To specify whether analysis is coupled or purely electrical

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: None

PCOM SCOM input parameters

CUPL *copt*

copt (C) Coupling option. Enter OFF or ELEC for electric solve only. Default = ON; Both electrialc and mechanical..

Notes: 1. If mechanical is active, the maximum timestep is determined by the stability criterion for mechanical wave propagation. If electrical only, the TIME command can be used to set a suitably accurate timestep.

ECON (Electrode Interconnect) Subcommand
PIEZ Input Group **PIEZ-ECON**

Purpose: To define lumped parameter elements connecting electrodes. Resistors, capacitors, inductors are currently implemented.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **PIEZ** Group: Must follow **DEFN** subcommand which initated the electrodes.

PCOM SCOM input parameters

ECON *electrode1 electrode2 type value(s)*

electrode1, electrode2 = (C) electrodes to be connected

type = (C) Type of circuit component.
Any of: REST = resistor
CPAC = capacitor
INDR = inductor
SRLC = series RLC compound element
PRLC = parallel RLC compound element

values = (F) One or more values defining the properties of this component. See below for specifics.

ECON REST *rval* for a resistor

rval= (F) Resistance. Default = 0.0

ECON CPAC *cval* for a capacitor

cval= (F) Capacitance. Default = 0.0

ECON INDR *lval* for an inductor

lval= (F) Inductance. Default = 0.0

(description of **ECON** subcommand continues on next page)

ELEM(Element) Subcommand
CIRC (Circuit) Input Group **CIRC-ELEM**

(continued)

PCOM SCOM input parameters

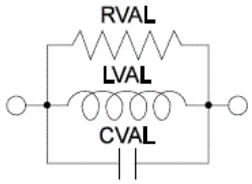
ECON SRLC *rval lval cval* for a series RLC element

rval= (F) Series inductance Default = 0.0, see note 1 below.

lval= (F) Series inductance Default = 0.0

cval= (F) Series capacitance Default = 0.0

ECON PRLC *rval lval cval* for a parallel RLC element



rval= (F) Parallel inductance Default = 0.0, see note 2 below.

lval= (F) Parallel inductance Default = 0.0

cval= (F) Parallel capacitance Default = 0.0

Notes:

1. The SRLC compound element consists of a resistor, an inductor and a capacitor connected in series. Setting *cval* = 0 or default results in infinite capacitance (ie, no capacitor).
2. Likewise, the PRLC compound element consists of a resistor, an inductor and a capacitor connected in parallel. Setting *rval* = 0 or default results in infinite resistance (ie, no resistor). Setting *lval* = 0 results in infinite inductance (ie, no inductor).

PLOD (pressure load) Input Command

Purpose: To provide the option of applying pressure loads on any surface of any continuum element in the grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the **GRID** command and any **DATA HIST** commands referred to by this command group.

PCOM	SCOM	input parameters
PLOD		
	PDEF	*plodname presname (chargenum) presscale timeshift timescale*
	SWEP	*plodname presname presscale timeshift timescale rateswep vxswep & vyswep vzswep xrefswep yrefswep zrefswep*
	SPOT	*spotname sx sy sz*
	VCTR	*vectorname vx vy vz*
	MATR	*mname material option*
	SDEF	*plodname direcname ibegin iend jbegin jend kbegin kend*
	SDEF	*plodname direcname* (LINE *width*) *i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
	SDF2	*plodname direcname matname1 matnam2 ibegin iend jbegin jend kbegin kend*
	CYLN	*plodname direcname (axisname) iaxis cbegin cend center1 center2 & radius thetabeg thetaend ibegin iend jbegin jend kbegin kend*
	SPHR	*plodname direcname xcent ycent zcent radius thetabeg thetaend & phibegin phiend ibegin iend jbegin jend kbegin kend*
	SNGL	*plodname direcname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
	GCON	*plodname direcname nodegc1 nodegc2 nodegc3 nodegc4*
	LDEF	*option*
	CHEK	*overlapcheck voidcheck*
	DACT	*plodname option*
	PRNT	*option*
	END	

Note: PLOD quantities may be visualized via the **GRPH** subcommands: **PLOT** PLOD_PRES and for conwep functions: **PLOT** PLOD_RANG.

PDEF (pressure definition) Subcommand
PLOD (pressure load) Input Group **PLOD-PDEF**

Purpose: To specify a pressure history to be used by the **PLOD** option.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must precede any **SDEF** subcommand that refers to the *plodname* defined by this subcommand.

PCOM	SCOM	input parameters
	PDEF	*plodname presname (chargenum) presscale timeshift timescale*

plodname = (C) Unique name assigned to this pressure load definition.

presname = (C) Name of the pressure-time history to assign to this pressure load definition. Set to ABLS if the **ABLS** command is used to define the pressure-time history or the water load name if the **WATR** command has been used. Set to ABNC if the **ABLS-ABNC** subcommand is used. Set to FUNC if the **FUNC** command is used to define the pressure-time history. Set to a *volumename* if a previously defined **CALC VOLM** and a volume-pressure relationship is used to define the pressure. Set to name of time history if the **DATA HIST** command is used to define a digitized time history. If a set of pressure-time histories contained within a single data array has been created and imported into the program, *presname* should identify the data array and the pressure record number within that array using the syntax *dname>n*. See Note 3 below.
No default.

chargenum = (I) Optional parameter used only if *presname* is set to ABNC. The ABNC charge number. ABNC charges are numbered in the order they are defined. Default = 1

presscale = (F) Scale factor by which the *presname* pressure values are multiplied. Default = 1.0

timeshift = (F) Shift value to add to the *presname* time values.
Default = 0.0

timescale = (F) Scale factor by which the *presname* time values are multiplied.
Default = 1.0

Notes: 1. The time values of the applied pressure-time history, t' , for this pressure load definition are defined relative to the original time values of *presname* , t , as:

$$t' = timeshift + (t \times timescale)$$

2. The starting pressure value of an applied pressure-time history must be consistent with the initial state of the model. Therefore, if the model is initially at rest, the prescribed pressure at the problem start time (typically 0.0) should be specified as 0.
3. Users may create a single 2D data array that contains a large number of pressure-time histories to be applied in an analysis as long as each pressure record has the same time sampling. This array is typically created using some other utility program or **Review**. The array must be 2D, i.e., PDAT(*ns,nt*), where *ns* defines the total number or pressure records plus one time record. The time record must be the first record in the array. The second subscript range, *nt*, represents the number of time points for a pressure record. This data array may be imported into **PZFlex** using a **DATA IN** command.
   Example:

```
DATA
        FILE IN MY.PRESSURE.DATA.FILE
        IN PDAT
        END
MGR PDAT    /* print the data array configuration in the Data Manager
C note that, if the PDAT is not properly dimensioned after import,
C use the DATA MGR command to set the proper 2D dimensions for this array
PLOD
        PDEF C1 PDAT>5 1.0 0.0 1.0 /* use the fifth pressure record in PDAT
        PDEF C2 PDAT>9 1.0 0.0 1.0 /* use the ninth pressure record in PDAT
        (etc.)
```

SWEP (sweep) Subcommand
PLOD (pressure load) Input Group **PLOD-SWEP**

Purpose: To specify a sweeping pressure history to be used by **PLOD**.

Use Is: Optional. If not input, a **PDEF** subcommand is required.

Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must precede any **SDEF** subcommand that refers to the *plodname* defined by this subcommand.

PCOM SCOM input parameters

SWEP *plodname presname presscale timeshift timescale rateswep vxswep & vyswep vzswep xrefswep yrefswep zrefswep*

plodname= (C) Unique name assigned to this sweeping load definition. No default.

presname = (C) Name of pressure-time history to assign to this definition. Set to FUNC if the **FUNC** command is used to define the pressure-time history. Set to name of time history if the **DATA HIST** command is used to define a digitized time history. No default.

presscale= (F) Scale factor by which presname pressure values are multiplied. Default = 1.0.

timeshift= (F) Time at which pressure wave arrives at (*xrefswep, yrefswep, zrefswep*) Default = 0.

timescale= (F) Scale factor by which the *presname* time values are multiplied Default = 1.0.

rateswep (F) Velocity at which the pressure load travels. Default = 0.

vxswep, vyswep,vzswep= (F) Direction vector of the travelling pressure load. No default.

xrefswep, yrefswep,zrefswep= (F) Reference point for sweeping pressure load. Default = (0.,0.,0.)

Note: The time values of the applied pressure history, t', for this load definition are defined relative to the original time values of presname, t, as:
*t' = timeshift + (t * timescale) + R / rateswep*
where the projected distance *R* is given by:
*R = (vx,vy,vz) * [(x-xrefswep), (y-yrefswep), (z-zrefswep)]*, and *(vx,vy,vz)* is a unit vector in the *(vxswep, vyswep,vzswep)* direction

SPOT Subcommand
PLOD (pressure load) Input Group **PLOD-SPOT**

Purpose: Specify a spot defining the side of a surface to which a normal pressure loading will be applied. A spot is a point in space which, if it were a light source, would illuminate the side of the elements to which pressure values will be applied.

Use Is: Optional. The **VCTR** subcommand provides an alternative way to apply a non-normal pressure loading.

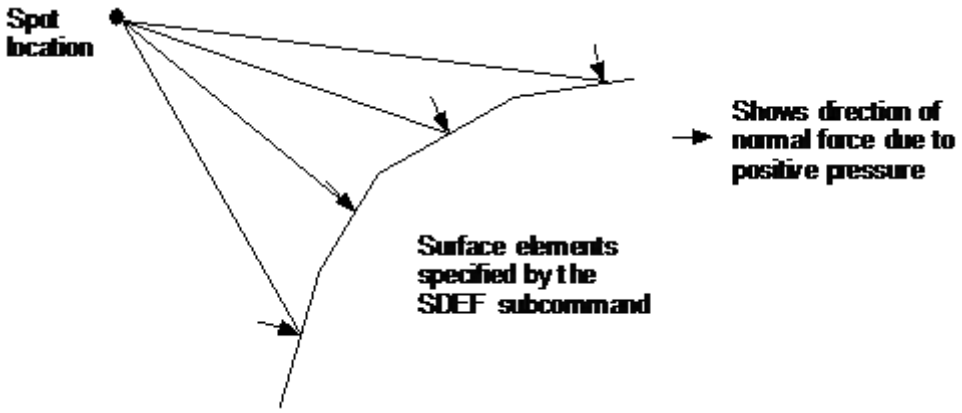
Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must precede any **SDEF** subcommand that refers to the *spotname* defined by this subcommand.

PCOM	SCOM	input parameters
	SPOT	*spotname sx sy sz*

spotname = (C) Unique name assigned to this spot definition. No default.

sx, sy, sz = (F) x, y, and z coordinates of the spot location. Default values = 0.0



SPOT subcommand illuminates the side of an element to which pressure is applied

Note: The *spotname* must be unique within the group of *spotname* and *vectorname* identifiers defined in the **PLOD** command group.

VCTR (vector) Subcommand
PLOD (pressure load) Input Group **PLOD-VCTR**

Purpose: To specify the direction of the force vector that results from the application of a positive pressure loading. The force direction is arbitrary, allowing the user to apply non-normal pressure loadings (i.e., shear traction, etc.). The direction of loading, as well as a pressure-time history, is associated with element surfaces by using the **SDEF** subcommand.

Use Is: Optional. The **SPOT** subcommand is another alternative method to specifying the force direction for positive pressure values if the pressure loading is normal to the surface.

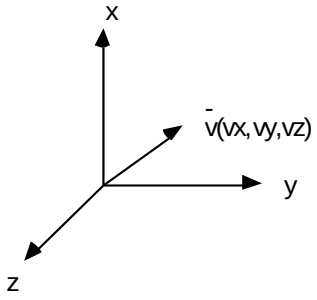
Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must precede any **SDEF** subcommand that refers to the *vectorname* defined by this subcommand.

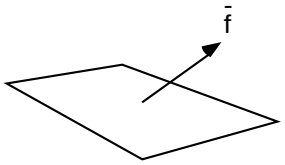
PCOM	SCOM	input parameters
	VCTR	*vectorname vx vy vz*

vectorname = (C) Unique name assigned to this vector definition. No default.

vx, vy, vz = (F) x, y, and z components defining the direction of the positive force vector. No default.



a. Vector defined with VCTR subcommand



b. Force due to +pressure acts in the v direction.

Using the **VCTR** subcommand to define the direction of a positive pressure force vector

Notes: 1. The parameter *vectorname* must be unique within the group of *spotname* and *vectorname* identifiers defined in the **PLOD** command group.
2. The magnitude of the force vector at any time is equal to the applied pressure at that time, multiplied by the surface area of the element being loaded. The input **VCTR** components are normalized to 1.0 and thus carry no magnitude information.

MATR (Material) Subcommand
PLOD (Pressure load) Input Group **PLOD-MATR**

Purpose: To specify which side of a surface is loaded via material name. This is an alternate to the SPOT or VCTR subcommands. The direction of loading, as well as a pressure-time history, is associated with element surfaces by using the **SDF2** subcommand.

Use Is: Optional. The **SPOT and VCTR** subcommands are alternatives to specifying the force direction.

Multiple Use: Allowed.

Order Dependence
Within **PLOD** Group: Must precede any **SDF2** subcommand which refers to the *vectorname* defined by this subcommand.

PCOM SCOM input parameters

MATR *mname materialname option*

mname = (C) A unique name assigned to this material definition. No default.

materialname = (C) The material name which defines the loaded side of a material interface. No default.

option = (C) IN or OUT Default = OUT. OUT means that the PLOD is normal to the interface, directed away from *materialname*

Notes: 1. The parameter *mname* must be unique within the group of *mname*, *spotname* and *vectorname* identifiers defined in the **PLOD** command group.

SDEF (surface definition) Subcommand
PLOD (pressure load) Input Group **PLOD-SDEF**

Purpose: To specify a surface to which a pressure load is applied.
Use Is: Required.
Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must follow any **PDEF, SWEP, SPOT** or **VCTR** subcommands which this subcommand references.

PCOM SCOM — input parameters

SDEF *plodname direcname* (LINE *width*) *ibegin iend jbegin jend kbegin kend*

plodname = (C) Name of the pressure load to be applied.
No default.

direcname = (C) Appropriate *spotname* or *vectorname* identifier which defines the direction of the of the force vector for a positive pressure load. For the LINE option, this must be a *vectorname*. No default unless this is an ABNC pressure definition. See Note 3 below.

(optional input to define a line load on a single row of nodes instead of on a 2D surface)

LINE *width* = (F) width of a pressure-loaded surface for line load. Must be preceded by the LINE indicator.
Default = 1.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the loaded surface.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the loaded surface.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the loaded surface.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. This subcommand defines a single surface, therefore *ibegin* must equal *iend,* or *jbegin* must equal *jend,* or *kbegin* must equal *kend.*
2. When the **SPOT** subcommand is used to apply normal pressure loading, the surface normal is computed for each element as the cross product of the diagonals of the element.
3. For surfaces loaded with the ABLS-ABNC command, the default for *direcname* results in the charge location being used as the spot location.

SDF2 (Surface definition 2) Subcommand
PLOD (pressure load) Input Group **PLOD-SDF2**

Purpose: To specify a surface to which a pressure load is applied. The surface is defined as the interface between two material regions of the model.

Use Is: Optional. Either **SDEF** or **SDF2** subcommands, however, must be input to define pressure loading.

Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must follow any **PDEF, SWEP, SPOT,** or **VCTR** subcommands that this subcommand references.

PCOM SCOM input parameters

SDF2 *plodname direcname matname1 matnam2 ibegin iend jbegin jend & kbegin kend*

or

SDF2 *plodname direcname matname1* **side** *iside ibegin iend jbegin jend & kbegin kend*

plodname = (C) Name of the pressure load to be applied.
No default.

direcname = (C) Appropriate *spotname* or *vectorname* identifier that defines the direction of the of the force vector for a positive pressure load. For the LINE option, this must be a *vectorname*.
No default.

matname1
matname2 = (C) Names of the two continuum materials whose interface defines the surface to be loaded. Set *matname2* to ALL to load all interfaces with *matname1*
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the grid region that will be searched for the material interface.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the grid region that will be searched for the material interface.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the grid region that will be searched for the material interface.
Default: *kbegin* = 1, *kend* = maximum K

iside = (I) If keyword **side** has been entered, then nodes on *iside* of the model in contact with *matname1* are added to the plod surface definition. No default.

Notes:
1. This option is useful for loading non-Cartesian surfaces in a standard partition grid.
2. When the **SPOT** subcommand is used to apply normal pressure loading, the surface normal is computed for each element as the cross product of the diagonals of the element.
3. Dummy materials, with zero density and stiffness can be defined and assigned to grid elements as an aid in "defining" the surface to be loaded.

CYLN (Cylinder) Subcommand
PLOD (Pressure load) Input Group **PLOD-CYLN**

Purpose: To specify a cylindrical surface to which a pressure load is applied.

Use Is: Optional. However, either **SDEF** or **SDF2, SPHR or CYLN** subcommands must be input to define pressure loading.

Multiple Use: Allowed.

Order Dependence
Within **PLOD** Group: Must follow any **PDEF, SWEP, SPOT** or **VCTR** subcommands which this subcommand references.

PCOM SCOM input parameters

CYLN *plodname direcname (axisname) iaxis cbegin cend center1 center2 & radius thetabeg thetaend ibegin iend jbegin jend kbegin kend*

plodname = (C) Name of the pressure load to be applied.
No default.

direcname = (C) The appropriate *spotname* or *vectorname* identifier which defines the direction of the of the force vector for a positive pressure load. No default.

axisname = (C) Optional input parameter. If the orientation of the cylinder axis does not align with the standard global axes of the model (STND), then input *axisname* as the name of the local coordinate system for this cylinder. This axis definition must have been previously defined by the **AXIS** command group.
Default = STND

iaxis = (C) Required if *axisname* is specified, optional if *axisname* is defaulted. *iaxis* defines the longitudinal coordinate axis of the cylinder.
Any of: X, Y or Z. Default = Z

cbegin, cend = (F) The beginning and ending *iaxis* coordinate values along the length of the cylindrical region.
No default.

center1, center2 = (F) The coordinates of the regions center axis.
Default = 0.0
If *iaxis* = X:
center1 = y-coordinate, *center2* = z-coordinate
If *iaxis* = Y:
center1 = z-coordinate, *center2* = x-coordinate
If *iaxis* = Z:
center1 = x-coordinate, *center2* = y-coordinate

CYLN (Cylinder) Subcommand
PLOD (Pressure load) Input Group

PCOM SCOM input parameters

radius= (F) The radius of the cylindrical region
Default = *0*.

thetabegin = (F) The beginning angle of the sector [degrees]. Theta is measured from x if iaxis = z, from y if iaxis = x, and from z if iaxis = y.
Default =0.

thetaend = (F) The ending angle of the sector [degrees].
Default = 360.

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the grid region that will be searched for the cylinder interface.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the grid region that will be searched for the cylinder interface.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the grid region that will be searched for the cylinder interface.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. This option is useful for loading non-Cartesian surfaces in a standard partition grid.
2. When using the **SPOT** subcommand to apply normal pressure loading, the surface normal is computed for each element as the cross product of the diagonals of the element.

SPHR (Sphere) Subcommand
PLOD (Pressure load) Input Group **PLOD-SPHR**

Purpose: To specify a spherical surface to which a pressure load is applied.

Use Is: Optional. However, either **SDEF** or **SDF2, SPHR or CYLN** subcommands must be input to define pressure loading.

Multiple Use: Allowed.

Order Dependence
Within **PLOD** Group: Must follow any **PDEF, SWEP, SPOT** or **VCTR** subcommands which this subcommand references.

PCOM SCOM input parameters

SPHR *plodname direcname xcent ycent zcent radius thetabeg thetaend & phibegin phiend ibegin iend jbegin jend kbegin kend*

plodname = (C) Name of the pressure load to be applied. No default.

direcname = (C) The appropriate *spotname* or *vectorname* identifier which defines the direction of the of the force vector for a positive pressure load. No default.

xcent, ycent, zcent = (F) Coordinates of the sphere's center. Default = 0.

radius= (F) The radius of the cylindrical region Default = *0*.

thetabegin = (F) The beginning angle of the sector [degrees]. Theta is measured from z (see AXIS command for a sketch).

thetaend = (F) The ending angle of the sector [degrees]. Default = 180.

phibegin = (F) The beginning angle of the sector [degrees]. Phi is measured from x (see AXIS command for a sketch).

phiend = (F) The ending angle of the sector [degrees]. Default = 360.

(continued next page)

PCOM SCOM input parameters

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the grid region that will be searched for the spherical interface. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the grid region that will be searched for the spherical interface. Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the grid region that will be searched for the spherical interface. Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. This option is useful for loading non-Cartesian surfaces in a standard partition grid.
2. When using the **SPOT** subcommand to apply normal pressure loading, the surface normal is computed for each element as the cross product of the diagonals of the element.

SNGL (single) Subcommand
PLOD (pressure load) Input Group **PLOD-SNGL**

Purpose: To specify a single (arbitrary) quadrilateral surface to which a pressure load is applied.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must follow any **PDEF, SWEP, SPOT,** or **VCTR** subcommands that this subcommand references.

PCOM SCOM input parameters

For 3D Models

SNGL *plodname direcname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*

or

SNGL *plodname direcname* LINE *width i1 j1 k1 i2 j2 k2*

For 2D Models

SNGL *plodname direcname i1 j1 i2 j2*

plodname = (C) Name of the pressure load to be applied. No default.

direcname = (C) The appropriate *spotname* or *vectorname* identifier that defines the direction of the of the force vector for a positive pressure load. For the LINE option, this must be a *vectorname.* No default.

(optional input to define a line load on a single row of nodes instead of on a 2D surface)

LINE *width* = (F) width of a pressure-loaded surface for line load. Must be preceded by the LINE indicator. Default = 1.

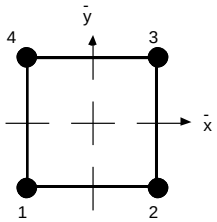
i1,j1,k1 = (I) ijk indicies for node 1 of the loaded quadrilateral. No Default.

i2,j2,k2 = (I) ijk indicies for node 2 of the loaded quadrilateral. No Default.

i3,j3,k3 = (I) ijk indicies for node 3 of the loaded quadrilateral. No Default.

i4,j4,k4 = (I) ijk indicies for node 4 of the loaded quadrilateral. No Default.

Note: The nodes should be ordered as shown below:



GCON (general connectivity) Subcommand
PLOD (pressure load) Input Group **PLOD-GCON**

Purpose: To specify a single (arbitrary) quadrilateral surface to which a pressure load is applied using general connectity syntax for node mapping.
See Section 2: "Importing General Connectivity Models"

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PLOD** Group: Must follow any **PDEF, SWEP, SPOT,** or **VCTR** subcommands that this subcommand references.

PCOM SCOM input parameters

For 3D Models

GCON *plodname direcname nodegc1 nodegc2 nodegc3 nodegc4*

or

GCON *plodname direcname* LINE *width nodegc1 nodegc2*

For 2D Models

GCON *plodname direcname nodegc1 nodegc2*

plodname = (C) Name of the pressure load to be applied. No default.

direcname = (C) Appropriate *spotname* or *vectorname* identifier that defines the direction of the of the force vector for a positive pressure load. For the LINE option, this must be a *vectorname*. No default.

(optional input to define a line load on a single row of nodes instead of on a 2D surface)

LINE *width* = (F) width of a pressure-loaded surface for line load. Must be preceded by the LINE indicator. Default = 1.

nodegc1, nodegc2
nodegc3, nodegc4 = (I) Local general connectivity node numbers defining the connectivity for this element. Note that triangular faces can be defined by setting *nodegc3*=*nodegc4*
No defaults..

LDEF (large deformation) Subcommand
PLOD (pressure load) Input Group **PLOD-LDEF**

Purpose: To control whether pressure loads are applied normal to the deformed surface for large deformation problems (i.e., follower loads). Applicable only for large deformation problems. If not input, the pressure load is always applied normal to the deformed geometry.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
within **PLOD** Group: None.

PCOM	SCOM	input parameters
	LDEF	*option*

option = (C) Large deformation option, either ON or OFF. Default is ON.

Notes: 1. If the **LDEF** subcommand is input for problems that are not large deformation (set with the **LDEF** primary command), no action is taken.
2. The **LDEF** subcommand has no effect on pressure loads that are applied using the **VCTR** subcommand to define the direction of loading.
3. Both surface rotation and change in area are taken into account if the **LDEF** *option* is ON.

CHEK (check) Subcommand
PLOD (pressure load) Input Group **PLOD-CHEK**

Purpose: To change the error checking rules for **PLOD** surface definitions.

Use Is: Optional. If not input, both the overlap and void node checks are on.

Multiple Use: Allowed.

Order Dependence
within **PLOD** Group: Error checking rules specified apply to all following **PLOD** subcommands.

PCOM SCOM input parameters

CHEK *overlapcheck voidcheck*

overlapcheck = (C) Check for overlapping specification of surface segments. Either ON or OFF.
If ON, a surface segment is not allowed to be loaded more that once.
Default = ON.

voidcheck = (C) Check for pressure loading on a surface segment having a void (massless) node (i.e., a nonphysical element).
ON, OFF, or DACT
ON = applying pressure to a surface segment containing a void node results in an error.
OFF = no checking is done, pressure may be applied to surface segments that contains void nodes. No warnings are issued
DACT = if a pressure is applied to a surface segment containing a massless node, it is automatically deactivated so that no pressure is applied to this surface segment.
Default = ON.

DACT (deactivate) Subcommand
PLOD (pressure load) Input Group **PLOD-DACT**

Purpose: To deactivate previously declared pressure loads.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PLOD** Group: This option can be input only after the PRCS step has been performed. Only previously defined pressure definitions can be deactivated.

PCOM SCOM input parameters

DACT *plodname option*

plodname = (C) Name of the plod definition (defined by the **PDEF** subcommand) to deactivate. All element faces loaded with this pressure definition are deactivated.
No default

option = (C) Option for treating the pressure that is currently applied as part of this plod definition
Either REMV or LEAV.
If REMV, the current pressure is removed and no more pressure loading is applied as part of this PLOD definition.
If LEAV, the current pressure remains as a continuiously applied pressure load that does not change during the rest of the simulation.
Default = REMV.

PRNT (print) Subcommand
PLOD (pressure load) Input Group **PLOD-PRNT**

Purpose: To print information about the PLOD command group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **PLOD** Group: None.

PCOM SCOM input parameters

PRNT *option*

option = (C) Option specifying which information to print
If SUMMARY, a summary of the current pressure loading is printed. If 'OFF' or 'ON' echo printing is controlled.
Default = SUMMARY

POUT (printed output) Input Command

POUT

Purpose: To provide a quick way to view output results without requiring additional postprocessing. This option allows the user to specify analysis results that are to be printed to the Job Output File during the execution phase. Although any snapshot data specified are printed only and not saved for future use, the time histories specified are also written to the Time History File at the end of the job in time history form.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede the **EXEC** commands that compute the desired results.

PCOM	SCOM	input parameters
POUT		
	FORM	*format idtag*
	HIST	*datanames ibegin iend iinc jbegin jend jinc kbegin kend kinc*
	HISTNAME	OBJECT_TYPE *option object_name*
	PLOT	*option*
	RATE	*iratehist iratesnap*
	SNAP	
PRNT		<----------standard **PRNT** primary command---------->
SHOW		<----------standard **SHOW** primary command---------->
	END	

Notes: 1. Once a **POUT** command group has been used to request the desired output, follow on **POUT** commands have the following actions depending upon which of its subcommands are used:

a. Used with only the **RATE** subcommand. Resets the rate for the previously requested time history and snapshot data.

b. Used with the **SNAP** subcommand. Purges any previously requested snapshot group. The newly defined snapshot information is active from this point on.

c. Used with the **HIST** subcommand (*not recommended*). Once a group of time histories has been requested with a group of **HIST** subcommands within a job, no new **HIST** commands should be specified in the same job. Using different **HIST** subcommands in later restarts is possible but complicates the process of combining the Time History Output Files created by the separate executions. If a new set of **HIST** subcommands is provided, the preceding set is purged and the newly requested time histories accumulated from that point on.

FORM Subcommand
POUT (Printed Output) Input Group **POUT-FORM**

Purpose: To set the format for writing time history output.

Use Is: Optional. If not input, *format* = FLEX option is assumed.

Multiple Use: Not Allowed.

Order Dependence
Within **POUT** Group: None.

PCOM SCOM input parameters

FORM *format idtag mode*

format = (C) Format of history file. Any of:

FLEX = write using flxhst format.
MATLAB = write using matlab .mat format

Default = FLEX

idtag = (C) Two letter identifier for histories written to matlab .mat files. Default = H1.

mode = (C) Any of w, w7.3. Default = w7.3.

Notes: 1. History vectors written to matlab .mat files have the two letter *idtag* prepended to the vector name.
2. w = MATLAB 7.0 format; w7.3 = Creates a MAT-file in an HDF5-based format that can store objects that occupy more than 2 GB
3. Example input for writing matlab .mat file history files:

```
POUT
    FORM matlab h1
    RATE 2
    HIST xdsp 1 2 1  1 2 1  1 1 1
    END
```

HIST (history) Subcommand
POUT (printed output) Input Group **POUT-HIST**

Purpose: To define a group of time histories to be saved in memory during an analysis, printer-plotted in the Job Output File (unless suppressed by the **PLOT** subcommand), and saved at the end of the job on the Time History Output File. How often these time histories are sampled is controlled by the **RATE** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **POUT** Group: None.

PCOM SCOM input parameters

HIST *datanames ibegin iend iinc jbegin jend jinc kbegin kend kinc*

or

HIST *xyz datanames xbegin xend xinc ybegin yend yinc zbegin zend zinc*

datanames = (C) Name or names of data arrays for which time histories are desired for the array indices provided. Up to 10 names can be placed before the indices definition is input. Data names include any of the Data Manager array (See user manual)
No default.

ibegin, iend, iinc = (I) Beginning and ending I-indices and indices increment for data array values for which time histories are required.
Default: *ibegin* = 1, *iend* = *ibegin*, *iinc* = 1

jbegin, jend, jinc = (I) Beginning and ending J-indices and indices increment for data array values for which time histories are required.
Default: *jbegin* = 1, *jend* = *jbegin*, *jinc* = 1

kbegin, kend, kinc = (I) Beginning and ending K-indices and indices increment for data array values for which time histories are required.
Default: *kbegin* = 1, *kend* = *kbegin*, *kinc* = 1

xbegin, xend, xinc = (F) Beginning and ending x-coordinates and x increment for data array values for which time histories are required.
Default: *xbegin=xend=xinc=0.*

ybegin, yend, yinc = (F) Beginning and ending y-coordinates and y increment for data array values for which time histories are required.
Default: *ybegin=yend=yinc=0*

zbegin, zend, zinc = (F) Beginning and ending z-coordinates and z increment for data array values for which time histories are required.
Default: *zbegin=zend=zinc=0.*

HISTNAME (history name) Subcommand
POUT (printed output) Input Group **POUT-HISTNAME**

Purpose: To make time requests for specific named objects in the model (specialized option). Unlike the **HIST** subcommand, allows the user to request time histories for any Data Manager Array location, **HISTNAME** is designed to use a "named object" syntax instead of array location indices.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **POUT** Group: None.

PCOM SCOM input parameters

< for electrode information>

HISTNAME ELECTRODE *option electrode_name*

option = (C) Character string that identifies the time history data that will be saved for this electrode. Any combination of the letters: V =voltage,
Q = charge
I = current
Default = VQ

electrode_name = (C) Name of the electrode. If set to ALL, then time histories will be saved for all electrodes.
Default: ALL

< for CALC AVRG information>

HISTNAME AVRG *option name*

option = (C) A character string that identifies the time history data that will be saved for this average. Any combination of the letters:
A = average,
S = sum
W = weights
Default = A

name = (C) The name of the average. If set to ALL, then timehistories will be saved for all averages.
Default: ALL

(Continued next page)

< for CALC BWORK information>

HISTNAME BWORK *option name*

option = (C) A character string that identifies the time history data that will be saved for this BWORK surface. Any combination of the letters:
W = WORK
P = POWER
Default = W

name = (C) The name of the BWORK surface. If set to ALL, then time histories will be saved for all BWORK surfaces.
No default

< for CALC VOLMINTG information>

HISTNAME VOLMINTG *option name*

option = (C) Set to *

name = (C) The name of the Volume integral. If set to ALL, then time histories will be saved for all Volume integrals.
No default

< for GAGE information>

HISTNAME GAGE *name*

name = (C) The GAGE label. If set to ALL, then time histories will be saved for all Gage points.
No default

(Continued next page)

POUT (Printed Output) Input Group **POUT-HISTNAME**
(continued)

< for CALC RIGD information>

HISTNAME RIGD *option1 option2 name*

option1 = (C) A character identifying the time history data type that will be saved for this rigid body. Either of:
F = forces (RGFR),
V = velocities (RGVL)
Default = F

option2 = (C) A character string that identifies the components of the data type specified in *option1* that will be saved as time histories. Any combination of the letters:
X = X-component,
Y = Y-component,
Z = Z-component,
RX = X-rotational component,
RY = Y-rotational component,
RZ = Z-rotational component,
A = All components
Default = A

name = (C) The name of the rigid body. If set to ALL, then time histories will be saved for all rigid bodies.
Default: ALL

<for Review SHAPE information>

Provides a convenient method to output velocity information to compute SHAPE in Review. Information is written to the *shphst* file to avoid overwhelming the *flxhst* file with too many records. Model coordinate data is written to a *flxshp* file. *Shphst* has the same file format as *flxhst; flxshp* has the same format as *flxdato*. Thus *shphst* and *flxshp* can be read be Review to compute SHAPE.

HISTNAME SHAPE *ibeg iend iinc jbeg jend jinc kbeg kend kinc*

ibeg..., kend = (I) Nodal range and increments for output.
Default = entire model.

HISTNAME SHAPD *array ibeg iend iinc jbeg jend jinc kbeg kend kinc*

array = (C) Data array for which SHAPE will be calculated in Review. May be used for as many arrays as desired (or none). HISTNAME SHAPD should follow HISTNAME SHAPE. No default.

ibeg..., kend = (I) Nodal range and increments for output.
Default = entire model.

Note: Example:

```
POUT
C   the following command (in lower case) request the charge, voltage and current
C   for the electrode named "top"
    HISTNAME ELECTRODE  QVI  TOP
```

PLOT Subcommand
POUT (printed output) Input Group **POUT-PLOT**

Purpose: To control the printing at the end of a job of time history information requested by the **POUT** command group.

Use Is: Optional. If not input, *option* = NONE option is assumed.

Multiple Use: Allowed.

Order Dependence within **POUT** Group: None.

PCOM	SCOM	input parameters
	PLOT	*option*

option = (C) Time history printed output option.
Any of:
YES = make simple printer plots at end of job.
NO = suppress the printer plots but print minimum and maximum values for each time history.
NONE = suppress all printing associated with the time histories.

Default = NONE

Note: Time histories are printer-plotted 4 to a page. Using the ON option can result in a large amount of printed output if many time histories are requested.

RATE Subcommand
POUT (printed output) Input Group **POUT-RATE**

Purpose: To specify the time step rate at which time history information and snapshot data are to be processed.

Use Is: Required if snapshot data is desired. Also, if the user wants time history sampling at a rate other than 1.

Multiple Use: Allowed.

Order Dependence
within **POUT** Group: None.

PCOM SCOM input parameters

RATE *iratehist iratesnap*

iratehist = (I) Time step sampling rate at which time history data is recorded. *iratehist* = 1 records data for every time step, *iratehist* = 2 records data every other time step, etc.
Default = value defined on previous **RATE** subcommand if any, otherwise 1.

iratesnap = (I) Snapshots of field variables are printed to the Job Output File every *iratesnap* time steps.
Default = value defined on previous **RATE** subcommand if any, otherwise 0.

SNAP (snapshot) Subcommand
POUT (printed output) Input Group **POUT-SNAP**

Purpose: To define snapshot information to be printed to the Job Output File during problem execution. How often these snapshots are printed is controlled by the **RATE** subcommand. *None* of the requested snapshot data are saved to an external output file during the analysis. Use **OUTP SNAP** or **DATA OUT** commands to save snapshot data for future use.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **POUT** Group: Must immediately precede the group of **PLOT**, **PRNT,** and **SHOW** commands wthat define the snapshot data to display.

PCOM	SCOM	input parameters
	SNAP	-none-
PLOT		<----------standard **PLOT** primary command---------->
PRNT		<----------standard **PRNT** primary command---------->
SHOW		<----------standard **SHOW** primary command---------->

Notes: 1. The action of this command is unusual, as the **PLOT**, **PRNT,** and **SHOW** commands, which immediately follow the **SNAP** subcommand, are internally stored by PZFlex for future use. They perform no action when they are read in. During the execution phase, these snapshot commands are inserted into the command input stream every *iratesnap* time step (specified on the **RATE** subcommand), at which time their actions are performed.

2. If the snapshot commands do not directly follow the **SNAP** subcommand, they are interpreted the same as any other primary command and their actions are performed immediately instead of being deferred.

PRCS (process) Input Command

PRCS

Purpose: To construct an internal formulation of the defined problem to allow efficient solution of the model's response.

Use Is: Optional. If not input before the first **EXEC** command, the **PRCS** command is automatically performed.

Multiple Use: Not allowed.

Order Dependence: Must follow all problem definition input commands (i.e., **GRID**, **GEOM**, **MATR**, **MODL**, **BOUN**, **ZONE**, etc.).

PCOM	input parameters
PRCS	-none-

Notes:
1. The **PRCS** command computes the allowable time step for each zone.
2. All active data arrays for field variables are opened at the end of the **PRCS** step.

PRNT (print) Input Command

PRNT

Purpose: To print the values of variables in data arrays controlled by the Data Manager.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the point in the job processing at which the data array has become active in the Data Manager. For most field variable arrays, this is after the **PRCS** command.

PCOM input parameters

PRNT *dataname option ibegin iend jbegin jend kbegin kend*

dataname = (C) Name of data array containing elemental or nodal variables to be printed. No default.
For example: XDSP, SGYY, etc.

option = (C) Defines the print option for this command.
Any of:
I = print J-K plane of values for each I-index
J = print I-K plane of values for each J-index
K = print I-J plane of values for each K-index
MNMX = print the min and max values for the specified region of the data group
MNMZ = same as MNMX, but excluding zero
Default = K

ibegin, iend = (I) Beginning and ending I-indices for the region of the data array to be printed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for the region of the data array to be printed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices for the region of the data array to be printed.
Default: *kbegin* = 1, *kend* = maximum K

Note: Use of **PRNT** commands in conjunction with the **POUT SNAP** command provides a simple method for printing grid values throughout an analysis.

Example: The following command prints a 2D array of values of the z component of the velocity field at every mesh for the plane of data having a J-index of 10:

PRNT ZVEL J * * 10 10 * *

REGRID Input Command

REGRID

Purpose: To revise the computational mesh during an analysis. Typically used when elements become too small or distorted. Currently this is a very limited option. It can be used to move nodes in void regions (e.g., for electrostatic analyses), but field variables are not adjusted to account for the node motion.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow **PRCS** command.

PCOM SCOM input parameters

REGRID

FILL *(axisname) fillindex ibegin iend jbegin jend kbegin kend icomp*

END

FILL Subcommand
REGRID Input Group **REGRID-FILL**

Purpose: To compute the nodal coordinates for nodes in the interior of a bounded region by interpolation based on the coordinates of the bounding nodes of the region.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any command that references the coordinates of the nodes being assigned with this statement.

PCOM SCOM input parameters

<for 2D models>
FILL (*axisname*) *fillindex ibegin iend jbegin jend icomp*

<for 3D models>
FILL (*axisname*) *fillindex ibegin iend jbegin jend kbegin kend icomp*

axisname = (C) Optional input parameter. The name of a local axis system previously defined by the **AXIS** command group. This option is used only if the user would like to perform the fill operation in a cylindrical or spherical coordinate system causing the interpolation to work directly with the curvilinear coordinates of the local system. This produces filled grid lines that are curved instead of straight. If not input, the fill operation will work in the global Cartesian system.

fillindex = (C) Grid index fill parameter.

If a single index, any of: I, J, or K, then this defines the index direction to fill along.
For example, if *fillindex*=I, the coordinates of all nodes within the ijk window having *ibegin*< I-index < *iend* are defined by using equal-spaced interpolation based on the coordinates of the bounding nodes with
I-index=*ibegin* and I-index=*iend*.

If any combination of I, J, and K, such as JK, the program fills in the direction of the first index specified (J) but uses the coordinate positions on the bounding edges for the second index (K) in order to define the spacing along the first index. This allows for filling with non-uniform spacing. This option requires that the coordinates for the bounding edges of the two fill indexes be defined prior to using this command. No default.

ibegin, iend = (I) Beginning and ending I-indices of the bounding nodes for the region of nodes to be filled.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the bounding nodes for the region of nodes to be filled.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of the bounding nodes for the region of nodes to be filled.
Default: *kbegin* = 1, *kend* = maximum K

icomp = (C) Optional parameter. Used to perform the fill operation on only a single component of the coordinates for the nodes in the fill region. The other two coordinate components are left unchanged.
Any of: X, Y, or Z.
Default: all coordinate components are set with this command.

REST (restart) Input Command

REST

Purpose: To restart a job or control the writing of restart files.

Use Is: Required to resume processing of a previous job.

Multiple Use: Only one restart can be performed during any single execution of the program. The other functions of the **REST** command, however, may be used as often as desired in a run.

Order Dependence: Should be first input command of a restart run. May be placed anywhere if suppressing the writing of restart files.

PCOM input parameters

(define the name of the Restart Input File) **REST** FILE *restart_input_file*

(restart using the restart data set contained on the Restart Input File) **REST**

(write a Restart Output File at the present point in an analysis to a user's chosen file name) **REST** WRIT *restart_output_file*

(do not write a restart file of the type defined by *file_option* at the end of a run) **REST** NO *file_option*

(write a restart file of the type defined by *file_option* at the end of a run) **REST** YES *file_option*

restart_input_file = (C) Filename of the restart file to use when restarting a computation. No default

restart_output_file = (C) Filename that contains the restart data wittten by this command. This file should be different from the jobs standard Restart Output File. No default

file_option = (C) Type of restart files that is either written or suppressed by the **REST** NO/YES commands.
Any of: ALL, ERROR, NORMAL
If ALL, both normal and error restart files are affected
If NORMAL, only restart files produced with normal termination are affected.
If ERROR, only restart files written due to job termination as a result of an error are affected.
Default = ALL

Notes: 1. Once a job has been restarted, further processing of the job proceeds as though no break in job processing had occurred.

Examples:

```
C  ( standard command sequence to restart a job)
REST  FILE  flxrsto.run1    /* define name of Restart Input File
REST  /* restarts job by reading the first image from the Restart Input File

C  (suppresses the writing of any restart file by this job)
REST  NO

C  (suppress writing of restart if job terminates normally; write a restart if an error occurs)
REST  NO  NORMAL
```

2. Interim restart files may be written during a computation. The rate (time step interval) at which restarts are written is controled by the **OUTP RATE** command.
3. By default, no restart is produced for any run that has not completed the **PRCS** command. For a normal completion restart or error termination restart generated at the end of a run that ends prior to completing the **PRCS** command, use the **REST** YES option to turn on the saving of restarts at any point in the command sequence.

RIGD (rigid) Input Command

Purpose: To define a rigid substructure (a group of nodes that respond according to rigid body dynamics).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow **GRID** and precede the **PRCS** command.

PCOM SCOM — input parameters

RIGD

DEFN	*rigidname (axisname) xtmass ytmass ztmass xrmass yrmass zrmass xyrmass & yzrmass xzrmass*
CG	*xcg ycg zcg*
NODE	*ibegin iend jbegin jend kbegin kend (matname) (matnam2*
GCON	*nodegc1 nodegc2 nodegc3 ... nodegcn*
FIX	*dof(1) dof(2) ... dof(6)*
FREE	*dof(1) dof(2) ... dof(6)*
BC	*bctype histname scalx scaly scalz scalrx scalry scalrz* or *bctype histx histy histz histrx histry histrz*
SLAV	*inode jnode knode*
CHNG	*rigidname option* [parameters]
COMB	*maxloop*

Notes: 1. The global forces on and velocities of all rigid substructures are stored in the RGFR(i,j) and RGVL(i,j) data arrays, respectively. The units of the rotational velocities are radians/unit time. The j index of these arrays ranges from 1 to the number of rigid substructures in the order input. The I index ranges from 1 to 6 where:

- i = 1 is the x translational component
- i = 2 is the y translational component
- i = 3 is the z translational component
- i = 4 is the rotational component about x-axis
- i = 5 is the rotational component about y-axis
- i = 6 is the rotational component about z-axis

2. Nodes of rigid substructures are constrained to move according to rigid body dynamics unless the **FREE** subcommand is used to unconstrain certain degrees-of-freedom of the nodes. The rigid body behavior takes precedence over any other velocity boundary condition applied to a node assigned to the rigid body. Only bonded constraints take precedence over rigid body constraints. Users should avoid inconsistent rigid, bonded, and velocity constraints.

3. To apply initial velocities to the nodes of a rigid substructure requires setting the initial values in the RGVL array by using the **SET** command. These velocities are assigned to the nodes of the rigid substructure during the next computational time step. Note that RGVL is in global coordinates.

4. Substructures may be visualized via the graphics commands: PLOT RIGD, PLOT RIGD_XVEL, PLOT RIGD_RXVEL, etc. See also GRPH PSET BOXSIZE.

DEFN (definition) Subcommand
RIGD (rigid) Input Group **RIGD-DEFN**

Purpose: To begin the definition of a rigid substructure.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **RIGD** Group: Must precede any other subcommands that add to the definition of this rigid substructure.

PCOM	SCOM	input parameters
	DEFN	*rigidname (axisname) xtmass ytmass ztmass xrmass yrmass zrmass xyrmass & yzrmass xzrmass*

rigidname = (C) Unique name assigned to this rigid substructure.
No default.

axisname = (C) Optional input parameter. Name of previously defined local axis system.
Default = STND

xtmass, ytmass, ztmass = (F) Translational mass terms of the rigid substructure in the x-, y- and z- directions that are not included in the lumped continuum masses of the nodes of the rigid substructure.
Default values = 0.0

xrmass, yrmass, zrmass, xyrmass, yzrmass, xzrmass = (F) Rotational moment of inertia and product of inertia values for the rigid substructure about the x-, y-, and z-axes that are not included in the lumped continuum masses of the nodes of the rigid substructure. These values are applied at the center of gravity of the rigid substructure.
Default values = 0.0

Note: The complete mass characteristics of a rigid substructure are defined by the mass values assigned by the **DEFN** subcommand *plus* the lumped continuum masses at the nodes of the rigid substructure.

CG (center of gravity) Subcommand
RIGD (rigid) Input Group **RIGD-CG**

Purpose: To specify the center of gravity of the rigid substructure. If not included, the center of gravity is internally computed.

Use Is: Optional.

Multiple Use: Only one use is allowed for each rigid substructure.

Order Dependence
Within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	CG	*xcg ycg zcg*

xcg, ycg, zcg= (F) x-, y- and z-coordinates in the local coordinate system which define the center of gravity of the rigid substructure. Default values = 0.0

NODE Subcommand
RIGD (rigid) Input Group **RIGD-NODE**

Purpose: To define a group of grid nodes that are part of a rigid substructure.

Use Is: Optional. Either NODE or GCON subcommand must be input to define a rigid substructure..

Multiple Use: Allowed.

Order Dependence within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM SCOM input parameters

NODE *ibegin iend jbegin jend kbegin kend (matname) (matnam2)*

ibegin, iend = (I) Beginning and ending I-indices of a group of nodes which are part of the rigid substructure.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of a group of nodes that are part of the rigid substructure.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of a group of nodes that are part of the rigid substructure.
Default: *kbegin* = 1, *kend* = maximum K

matname = (C) Name of a previously defined material. If input, only nodes in contact with this material and within the ijk range are assigned to the rigid.

matnam2 = (C) Name of a previously defined material. If input, only nodes in on the *matname/matnam2* interface and within the ijk range are assigned to the rigid.

.

GCON (general connectivity) Subcommand
RIGD (rigid) Input Group **RIGD-GCON**

Purpose: To define a group of grid nodes which are part of a rigid substructure using general connectity syntax for node mapping. See Section 2: *Importing General Connectivity Models*

Use Is: Optional. Either GCON or NODE subcommand must be input to define a rigid substructure.

Multiple Use: Allowed.

Order Dependence
within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	GCON	*nocegc1 nodegc2 nodegc3 ... nodegcn*

nodegc1, nodegc2, ..., nodegcn = (I) List of nodes (local general connectivity node numbers) to assign to the rigid substructure.
No defaults.

FIX Subcommand
RIGD (rigid) Input Group **RIGD-FIX**

Purpose: To specify which local degrees-of-freedom of a rigid substructure are fixed, i.e., the velocity of these degrees of freedom of the rigid substructure are set to 0.0.

Use Is: Optional.

Multiple Use: Only one use is allowed for each rigid substructure.

Order Dependence
within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	FIX	*dof(1) dof(2) ... dof(6)*

dof(n) = (C) Local degrees of freedom of the rigid substructure which are to be fixed. Any of: X, Y, Z, RX, RY, RZ that are the three local translation and rotation degrees of freedom of the rigid substructure. Up to 6 dofs may be input.

Notes:
1. The **FIX** and the **BC** subcommands provide the only two ways to apply kinematic constraints to a rigid substructure, as kinematic boundary constraints defined by the **BOUN** command on nodes assigned to the rigid substructure are ignored when computing rigid substructure motions.
2. If a rigid substructure has no rotary inertia for one or more of its rotational degrees of freedom, these rotational degrees of freedom should be fixed.
3. The old form of this subcommand in which a 1 or -1 is placed in parameter locations 1 through 6 representing the different dofs of a rigid substructure is still supported. Use of the new form is recommended, however.

FREE Subcommand
RIGD (rigid) Input Group **RIGD-FREE**

Purpose: To decouple the motion of the specified degrees of freedom of nodes attached to the rigid substructure from the rigid body response of the rigid substructure. This option was previously part of the old form of the **FIX** subcommand.

Use Is: Optional.

Multiple Use: Only one use is allowed for each rigid substructure.

Order Dependence
within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	FREE	*dof(1) dof(2) ... dof(6)*

dof(n) = (C) Local degrees of freedom of the nodes attached to the rigid substructure that are to be decoupled from the rigid body response of the rigid substructure. Any of: X, Y, Z, RX, RY, RZ which are the three local translation and rotation degrees of freedom of the nodes. Up to 6 dofs may be input.
See Note 2 below.

Notes:
1. The motion of the rigid substructure is still defined by the summation of all forces acting on grid nodes assigned to the rigid substructure and any **BC** or **FIX** boundary constraints.
2. Note that freeing rotational degrees of freedom of nodes attached to a rigid substructure does not decouple the nodes entirely from the rotational motion of the rigid substructure, as any of the nodes' translational degrees of freedom still controlled by the rigid substructure may have a velocity contribution from the rigid substructure's rotational velocity. The rigid body's rotational degrees of freedom may be constrained (using the FIX subcommand) to avoid any influence of rigid body rotations on nodal motion.

BC (boundary condition) Subcommand
RIGD (rigid) Input Group **RIGD-BC**

Purpose: To prescribe a force- or velocity-time history condition for a rigid substructure in its local coordinate system.

Use Is: Optional.

Multiple Use: Only one use is allowed for each rigid substructure.

Order Dependence within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
BC		*bctype histname scalx scaly scalz scalrx scalry scalrz*
	or	
BC		*bctype histx histy histz histrx histry histrz*

bctype = (C) Type of boundary condition:
Any of: VEL = prescribes the velocity of the rigid substructure
VEL1 = similar to VEL, except that 0.0 values for any of the *scal* factors free that DOF of the rigid substructure from being affected by this velocity constraint.
FORC = adds a force contribution to the rigid substructure
No default.

histname = (C) Name of the time history to apply to the rigid substructure. Set to FUNC if the **FUNC** command is used to define the time history. Set to time history name if the **DATA HIST** command is used to define a digitized time history. No default.

scalx, scaly, scalz = (F) Scale factors for rigid local directions which, when multiplied by the time history values defined by *histname*, define the magnitude of force or velocity which will be used for each translational degree-of-freedom of the rigid substructure. Defaults = 0.0

scalrx, scalry, scalrz = (F) Scale factors for rigid local directions which, when multiplied by the time history values defined by *histname*, define the magnitude of force or velocity which will be used for each rotational degree of freedom of the rigid substructure. Defaults = 0.0

histx, histy, histz, histrx, histry, histrz = (C) Names of the time histories for each translational and rotational degree of freedom to apply to the rigid substructure. Set to the time history name of the **DATA HIST** command used to define the digitized time history. Default is constant 0.0. See note 1.

Note: If individual data time histories are entered for each rigid degree of freedom defaulting the input for a particular dof results in that dof being fixed for *bctype* = VEL and freed for *bctype* = VEL1.

SLAV (slave) Subcommand
RIGD (rigid) Input Group **RIGD-SLAV**

Purpose: To prescribe that a rigid substructure will be slaved to a node of the grid (i.e., the velocity of the rigid substructure and all grid nodes assigned to it are set equal to the velocity of the master grid node).

Use Is: Optional.

Multiple Use: Only one use is allowed for each rigid substructure.

Order Dependence within **RIGD** Group: Must follow the appropriate **DEFN** subcommand.

PCOM	SCOM	input parameters
	SLAV	*inode jnode knode*

inode, jnode, knode = (I) ijk-indices of the master node to which the rigid substructure is slaved.
Default values = 1

Notes:
1. The rotational velocities of a rigid substructure that has been slaved are set to 0.0.
2. The **SLAV** subcommand will override any other constraints specified for a rigid substructure including those requested by the **BC** and **FIX** subcommands.

CHNG (change) Subcommand
RIGD (rigid) Input Group **RIGD-CHNG**

Purpose: To change the constraints on a rigid substructure during execution.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **RIGD** Group: This option is allowed only after the **PRCS** command has been completed.

PCOM SCOM input parameters

CHNG *rigidname option* [parameters]

rigidname = (C) Name of the rigid substructure that this subcommand applies to.
No default.

option = (C) Type of rigid substructure attributes that are changed by this command.
No default.

For *option* = FIX, >> change the fixity definition of a rigid substructure
(replaces previous **RIGD FIX** command specification)

CHNG *rigidname* FIX *dof(1) dof(2) ... dof(6)*

dof(n) = (C) List of local degrees of freedom of the rigid substructurem to fix. Any of: X, Y, Z, RX, RY, RZ that are the three local translation and rotation degrees of freedom of the rigid substructure. Up to 6 dof's may be input. Any rigid substructure dof not included in this list is not fixed.

For *option* = FREE, >> change the specification of which nodal dofs are freed from the rigid substructure
(replaces previous **RIGD FREE** command specification)

CHNG *rigidname* FREE *dof(1) dof(2) ... dof(6)*

dof(n) = (C) List of local degrees of freedom of the nodes attached to the rigid substructure that are to be decoupled from the rigid body response of the rigid substructure. Any of: X, Y, Z, RX, RY, RZ that are the three local translation and rotation degrees of freedom of the nodes. Up to 6 dofs may be input. Any nodal dof not included in this list is constrained by the rigid substructure motion.

For *option* = BC, >> change the specification of boundary condition scale factore for individual dofs. (modifies previous **RIGD BC** command specification)

CHNG	*rigidname* BC *dof(1) scale(1) dof(2) scale(2) … dof(6) scale(6)*		
	dof(n) =	(C)	Local degrees of freedom of the rigid substructure to which the following BC scale factor applies. Any of: X, Y, Z, RX, RY, RZ that are the three local translation and rotation degrees of freedom of the nodes. Up to 6 dofs may be input. Any dof not defined with this command retains the scale factor specified on the previous **BC** subcommand for this rigid substructure.
	scale(n) =	(C)	New scale factor to use for the prescribed boundary condition function applied to *dof(n)*.

COMB (combine) Subcommand
RIGD (rigid) Input Group **RIGD-COMB**

Purpose: To combine rigid definitions that contain common nodes.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **RIGD** Group: Must follow the DEFN and NODE or GCON commands that define the rigid definitions to be combined.

PCOM SCOM input parameters

COMB *maxloop*

maxloop = (I) Maxiumum number of combination loops permitted
Default = 100.

Notes:
1. The command combines definitions that have common nodes, adding the nodes of the second definition to the first, merging duplicate nodes, keeping the name and properties of the first definition, and deleting the name and properties of the second.
2. The command reduces the number of rigid definitions in the model from what was input. Therefore subsequent commands that reference rigids need to be adjusted accordingly.
3. Important statistics on which definitions were combined, number of rigid nodes deleted, and number of rigid nodes and definitions remaining are printed out to the flxprt file.
4. The logic loops through all rigid definitions until either no common nodes remain or *maxloop* is reached.
5. Usually substructures with a common node should be defined as a single rigid. The unintentional definition of separate rigids with a common node might occur when small parts are made rigid to avoid timestep penalties associated with unimportant portions of a model. This option would be used to automatically combine those that should be combined.

SET Input Command

SET

Purpose: To set the value of any data array variables stored in the Data Manager.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the point in the job processing at which the data array has become active in the Data Manager. For most field variables, this is after the **PRCS** command.

PCOM — input parameters

SET *dataname value (add) ibegin iend jbegin jend kbegin kend*

dataname = (C) Name of data array containing variables to be set.
For example: XVEL, ZFRC, etc.
No default.

value = (F) Data value to assign to the specified variables.
Default = 0.0

(add) = (C) Optional input parameter. If set to ADD the data value is added to value already existing in the array location. Default is to overwrite the existing value.

ibegin, iend = (I) Beginning and ending I-indices for data array locations to be set to *value*.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for data array locations to be set to *value*.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices for data array locations to be set to *value*.
Default: *kbegin* = 1, *kend* = maximum K

Note: This command performs the following code:

```
      DO 10  K = kbegin, kend
      DO 10  J = jbegin, jend
      DO 10  I = ibegin, iend
      dataname(I,J,K) = value
10    CONTINUE
```

SHAP (shape) Input Command

Purpose: To extract steady-state deformation shapes at specified frequencies for regions of the model from a transient time domain analysis.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow the **GRID** command and precede **PRCS** command.

PCOM SCOM input parameters

SHAP

BASE	*arrayname ibase jbase kbase*
CNVRT	*option*
DATA	*arrayname*
FREQ	*frequency*
NODE	*ibegin iend jbegin jend kbegin kend*
TWND	*[timewindow] type option fraction*
END	

Notes:
1. The function of this command is to compute the Fourier transform of the nodal displacements for the requested nodes at the requested frequencies. The model needs to be excited by a driving signal that has significant frequency content in the frequencies specified by the **FREQ** subcommand and be run long enough to contain a reasonable number of periods for these frequencies.
2. The deformation shapes may be viewed with the SHAP option of the **GRPH PLOT** subcommand. This can be done in a restart job after the full set of time steps specified by the **TWND** subcommand has been executed or after the velocities of the nodes have decayed to zero.

BASE Subcommand
SHAP (Shape) Input Group **SHAP-BASE**

Purpose: To specify a baseline value to be subtracted from an array before computing SHAP.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SHAP** Group: None.

PCOM SCOM input parameters

BASE *arrayname ibase jbase kbase*

arrayname= (C) The arrayname to be modified. No default.

ibase = (I) *i* index of data value to be subtracted from arrayname before coimputing shapes. No default.

jbase = (I) *j* index of data value to be subtracted from arrayname before coimputing shapes. No default.

kbase = (I) *k* index of data value to be subtracted from arrayname before coimputing shapes. No default.

Notes: 1. To remove rigid body displacements from the shape plots, apply the BASE subcommand to xvel, yvel and zvel.
2. BASE can be applied to other data arrays to compute SHAP on relative values. Instead of computing shapes of *arraynam,e* BASE computes shape on the quantity *(arrayname(i,j,k) – arrayname(ibase,jbase,kbase))*

CNVRT (convert) Subcommand
SHAP (Shape) Input Group **SHAP-CNVRT**

Purpose: To convert the way shape geometry arrays are stored in the data manager between displacement, velocity and acceleration.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the **PRCS** command.

PCOM SCOM input parameters

CNVRT *option*

option = (C) Storage type. One of: DISP, VEL, ACCEL. Default = VEL.

Notes: 1. Shape geometry data is stored in arrays with the base names xvel, yvel and zvel in the shap directory of the data manager by default. If they are converted to DISP the base names become xvel-intg, yvel-intg and zvel-intg. If they are converted to ACCEL the base names become xvel-derv, yvel-derv and zvel-derv.
2. Converting between DISP, VEL and ACCEL does not change the way shape data is plotted.

DATA Subcommand
SHAP (shape) Input Group **SHAP-DATA**

Purpose: To specify additional data arrays for transform calculation.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **SHAP** Group: None.

PCOM SCOM input parameters

DATA *arrayname*

arrayname = (F) Name of the time vayring array, e.g., pres, for pressure field. No default.

FREQ (frequency) Subcommand
SHAP (shape) Input Group **SHAP-FREQ**

Purpose: To specify the frequencies at which deformation shapes are extracted.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHAP** Group: None.

PCOM SCOM input parameters

FREQ *frequency*

frequency = (F) Frequency in cycles/unit time.
No default.

NODE Subcommand
SHAP (shape) Input Group **SHAP-NODE**

Purpose: To specify a nodal region of the model where shape displacements will be computed.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHAP** Group: None.

PCOM SCOM input parameters

NODE *ibegin iend jbegin jend kbegin kend*

ibegin, iend = (I) Beginning and ending I-indices of a group of nodes for which the response shape is to be extracted.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend (I) Beginning and ending J-indices of a group of nodes for which the response shape is to be extracted.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of a group of nodes for which the response shape is to be extracted.
Default: *kbegin* = 1, *kend* = maximum K

TWND (time-window) Subcommand
SHAP (shape) Input Group **SHAP-TWND**

Purpose: To define a windowing function to scale a time history before computing deformation shapes. This option allows for computing useful results for data that are non zero at either end of the time history. It should be used if the transient analysis will not be run long enough to allow the response of the problem to decay to zero.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence within **SHAP** Group: None.

PCOM SCOM input parameters

TWND [*timewindow*] *type option fraction*

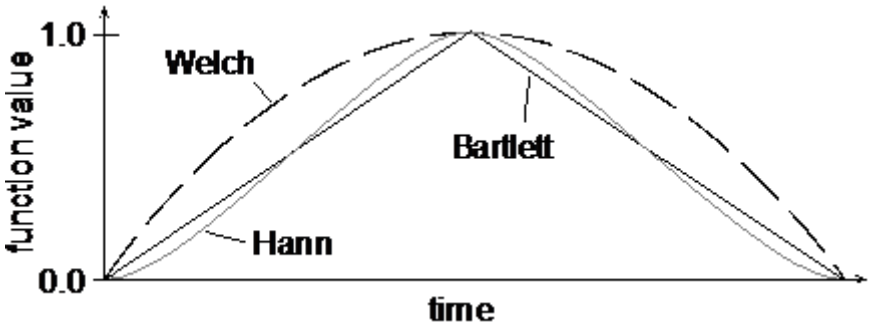
timewindow = (X) Time duration (defined as either the number of time steps or the simulation time) over which the window function is applied. The window function value is zero for all times greater than *timewindow*. If *timewindow* is an integer, it is assumed to be the number of time steps over which the window function should be applied. If it is input as a real number (i.e., with a decimal point), it is considered to be the simulation time over which the window function should be applied. No Default.

type = (C) Type of window function to use. See illustration.
Any of: BART = Bartlett window
WELC = Welch window
HANN = Hann window
Default = BART

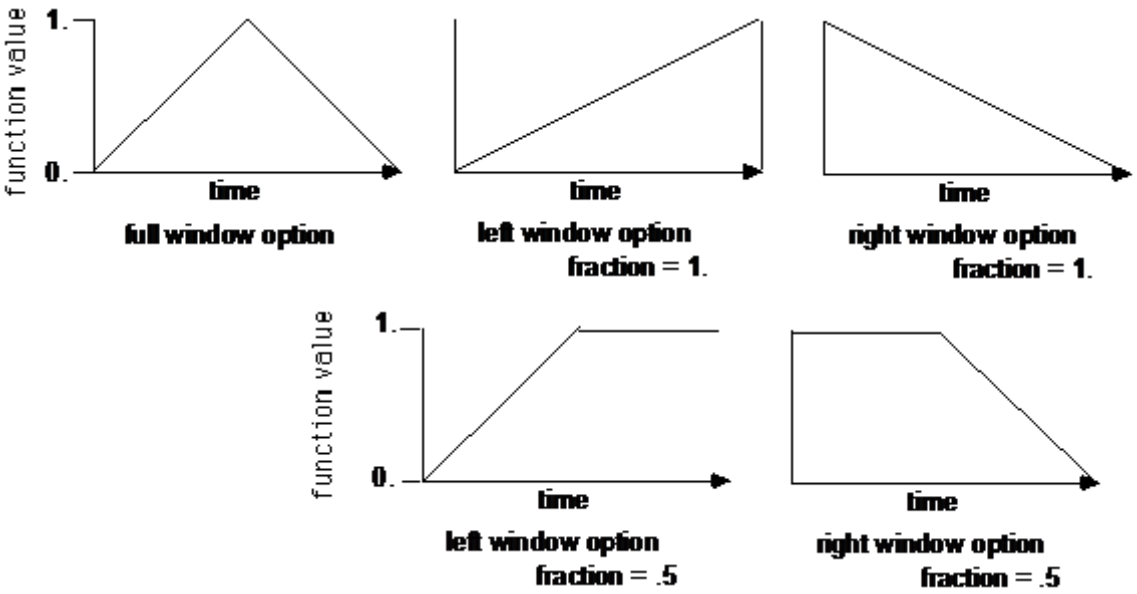
option = (C) Window option. See illustration.
Any of: FULL = Full window
LEFT = Left side of window
RIGH = Right side of window

Default = RIGH

fraction = (F) Fraction of a time history to apply the LEFT or RIGH window option to. Should be a value between 0. and 1.
Default = 1.0



Various windowing function types for the FULL window option



Various options for the Bartlett window function

Notes: 1. The function of this command is to scale the data values of a time history by the values of the window function at each corresponding time.

2. These window functions are discussed in *Numerical Recipes in FORTRAN, the Art of Scientific Computing - Second Edition* by W. H. Press, S. A. Teukolsky W. T. Vettering and B. P Flannery, Cambridge University Press, 1986.

SHEL (shell element) Input Command

Purpose: To define a bending plate or shell element for 2D and 3D models.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the **GRID** command and precede the **PRCS** command.

PCOM	SCOM	input parameters
SHEL		
	TYPE	*proptype*
	PROP	[*see specific model type for parameters*]
	THEX	*mname alphax alphay alphaz*
	AVIS	*mname* MATR *alinear aquadratic cmpopt strnrat*
	AVIS	*mname* CRIT *frequency fracdmp ellength*
	INFO	*option [parameters]*
	LAYR	*(option) lname thick nlayer pname(1) ...pname(nlayer) & hginpln hgnorm hgrot rotfac*
	ELEM	*lname ibegin iend jbegin jend kbegin kend (mat1) (mat2)*
	SNGL	*lname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
	GCON	*lname nodegc1 nodegc2 nodegc3 nodegc4 (groupnames)*
	DEL	*layrname ibeg iend jbeg jend kbeg kend*
	THIK	*lname fraction1 fraction2 ... fraction*
	OFFS	*lname zoffset*
	ANGL	*lname option angle1 angle2 ... anglen*
	BUFR	*nsiz_elem*
	LABL	*labelname ibegin iend jbegin jend kbegin kend touch_option*
	PRNT	*option*

Notes: 1. The material constitutive relations assume a von Mises yield criterion, with optional linear strain hardening, hardening limit, and viscoplastic effects.
2. The mass of shell elements are lumped to the element's nodes and the stability characteristics of shell elements are considered when computing zone time steps.

Notes: 3. Stresses for the element are stored in the SHSG(i,j) data array, and strains are stored in SHEP(i,j). For 2D, the i index of these arrays range from 1 to 3, where:

. i = 1 is the local xx component in the element
i = 2 is the local yy component in the element
i = 3 is the local xz component in the element

For 3D, the i index ranges from 1 to 5, where:

i = 1 is the local xx component in the element
i = 2 is the local yy component in the element
i = 3 is the local xy component in the element
i = 4 is the local xz component in the element
i = 5 is the local yz component in the element

Equivalent plastic strain and expended fraction of fracture energy are stored in SHPL(1,j) and SHPL(2,j), respectively. Fracture energy is calculated only when requested on the **PROP** subcommand. For layers with *proptype* = SFT1: SHPL(1,j), SHPL(2,j), and SHPL(3,j) contain EL, VDMG and EVBR respectively. For layers with *proptype* = PLIH, SHPL(1) contains equivalent plastic strain, and SHPL(2) contains expended fraction of post-peak plastic strain. For layers with *proptype*=ORTH: SHPL(1,j), SHPL(2,j), and SHPL(3,j) contain failure indicators, SHPL(1,j) contains the failure indicator for the material 1 direction (fiber), SHPL(2,j) contains the failure indicator for the material 2 direction (matrix), and SHPL(3,j) contains the failure indicator for shear. SHPL(i,j)≠0 indicates failure. For Yeoh materials, SHPL(1,j) contains the first deviatoric strain invariant. The j-index ranges from 1 to the number of layers for all elements in the order input. The **PRNT** subcommand provides a table relating layer numbers to shell elements.

4. The **GRPH PLOT** command documentation describes how to plot the local stress and strain components using names such as SG11 for the local xx stress, EP13 for the local xz strain, etc. The equivalent plastic strain for shell elements is named EPLS for plotting. The GRPH SET subcommand permits control over which layer to display.
5. The local coordinate system in which stress and strain quantities are defined depends on how each element is defined. The local system is shown on the appropriate **ELEM**, **SNGL,** or **GCON** subcommand page.

TYPE Subcommand
SHEL (shell) Input Group **SHEL-TYPE**

Purpose: To define the type of constitutive models that are defined by the following PROP subcommands.

Use Is: Optional. If not input, *proptype* = STND is default.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must precede **PROP** subcommands to which it applies.

PCOM SCOM input parameters

TYPE *proptype*

proptype = (I) The type of shell constitutive models that will be defined by following PROP subcommands.
Any of: STND or ORTH
STND = standard isotropic constitutive model or previously defined continuum SFT1, ELAS, VCAP or YEOH model
ORTH = orthotropic constitutive model

PROP (properties - STND) Subcommand
SHEL (shell) Input Group **SHEL-PROP**

Purpose: To define material properties for standard shell layers.

Use Is: Required if using standard shell constitutive models.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must precede any **LAYR** subcommand that references the defined material model.

PCOM SCOM input parameters

PROP *pname rho emodulus poisson sigy hmod siglim gfac (Keyword params) & edot(1) sig(1) edot(2) sig(2) ...edot(n) sig(n)*

pname = (C) Name of the shell layer material. No default.

rho = (F) Mass density of the material. No default.

emodulus = (F) Young's modulus of the material. No default.

poisson = (F) Poisson's ratio of the material. No default.

sigy = (F) Yield stress of the material in uniaxial stress test. *sigy* = 0.0 indicates a linear elastic material.
Default = 0.0

hmod = (F) Hardening modulus for the material in uniaxial stress test. *hmod* = 0.0 indicates a non-hardening material.
Default = 0.0

siglim = (F) Maximum stress after yield for the material.
For *siglim* < sigy, *siglim* is set to *sigy**10e6 (no limit).
Default = 0.0

gfac = (F) Transverse shear correction factor for the material. Default = 5/6 = 0.83333

The FRCT keyword and parameters are optional. If specified, they request a rate-independent, fracture energy-based softening. They preclude the specification of rate effects.

FRCT= (C) The characters FRCT (see notes 1 & 2 below)

epsoft= (F) The axial strain at which softening begins in unixial stress.

bsft= (F) Coefficient *b* in the softening function shown below.

The SFTN keyword and parameters are optional. If specified, they request a strain-based softening. Rate effects may be included.

SFTN= (C) Characters SFTN .

epsoft= (F) Axial strain at which softening begins in unixial stress.

asft= (F) Coefficient *a* in the softening function below.

bsft= (F) Coefficient *b* in the softening function below.

decay= (F) Exponent governing decay of rate effects with softening. Set to zero for no decay of rate effects. Nearest integer value is used. 0≤*decay*≤10. Default = 0.

The following parameters are for rate dependent behavior, default if no rate effects.

Up to 10 data pairs may be specified to define the sigmax vs. strain rate curve for uniaxial stress.

edot, sig= (F) Pairs (≤ 10) of plastic strain rate vs. axial stress in a uniaxial tension test. Leave blank for rate independent behavior. Shel prop input all goes on one line. Use the continuation character "&" as necessary.

Default = 0. - Rate independent

Notes: 1. The softening function used is:

$$\sigma_1=\sigma_{ult}\left[\frac{\varepsilon^{psft}}{\varepsilon^{p}}(1-a)+a\exp\left(-b\left(\varepsilon^{p}-\varepsilon^{psft}\right)\right)\right]$$

where,

$$\varepsilon^{psft}=\varepsilon^{sft}-\frac{\sigma_{ult}}{E}$$

and σ_{ult}=min(*sigultimate*, *sigyield+hardng*epsoft*), *a*=*asoft*, *b*=*bsoft*, σ_1=axial stress, E = *emodulus*, ε^{sft} = *epsoft*, and ε^{p} = plastic axial strain. The functional coefficients *a* and *b* are fit to the axial stress vs.. plastic strain behavior in uniaxial stress tests.

2. Entering FRCT triggers a fracture energy-based softening algorithm for which the stress-displacement response is independent of discretization. For this case, rate effects are not included. Softening follows the functional form of note 1 with *a* set to 1.0 and plastic strain replaced by displacement across an infinitesimal band.
3. The equivalent plastic strain is defined as:

$$d\bar{\varepsilon}=\sqrt{\frac{2de_{ij}^{p}de_{ij}^{p}}{3}}$$

PROP (properties - SFT1, PLIH or YEOH) Subcommand
SHEL (shell) Input Group **SHEL-PROP**

Purpose: To define SFT1 material properties for concrete shell layers, PLIH material properties for metals, or YEOH material properties for rubber-like shell layers. This option references previously defined continuum constitutive models.

Use Is: Required for using SFT1 shell constitutive models.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must precede any **LAYR** subcommand that references the defined material model.

PCOM SCOM input parameters

PROP *pname gfac*

pname = (C) Name of a previously defined SFT1, PLIH or YEOH constitutive model (defined in the MATR command group). No default.

gfac = (F) Transverse shear correction factor for the material. Default = 5/6 = 0.83333

Note: The PLIH model should not be used for shells if *dpflag* = off.

PROP (properties - ORTH) Subcommand
SHEL (shell) Input Group **SHEL-PROP**

Purpose: To define material properties for orthotropic shell layers.

Use Is: Required for using orthotropic shell constitutive models.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must precede any **LAYR** subcommand that references the defined material model.

PCOM SCOM input parameters

PROP *pname E1 E2 ν12 G12 G23 G31 sigyt1 sigyt2 tauy12 tauy23 tauy31 & gfac ncr sigyc1 sigyc2 tau*

pname = (C) Name of the shell layer material. No default.

rho = (F) Mass density of the material. No default.

E1 = (F) Young's modulus in x' direction of material. No default.

E2 = (F) Young's modulus in y' direction of material. No default.

ν12 = (F) Poisson's ratio in planeof material. No default.

G12 = (F) Shear modulus in x'-y' plane. No default.

G23 = (F) Shear modulus in y'-z' plane. No default.

G31 = (F) Shear modulus in z'-x' plane. No default.

gfac = (F) Transverse shear correction factor for the material. This value is used to scale the input values of G23 and G31. Default = 5/6 = 0.83333

<if all the following yield parameters are set to 0.0, the material is assumed to be elastic>

sigyt1 = (F) Tensile yield stress(strain) in the x' direction of the material. Default = 0.0

sigyt2 = (F) Tensile yield stress(strain) in the y' direction of the material. Default = 0.0

tauy12 = (F) Shear yield stress(strain) for shearing in x'-y' plane. Default = 0.0

tauy23 = (F) Shear yield stress(strain) for shearing in y'-z' plane. Default = 0.0

tauy31 = (F) Shear yield stress (strain) for shearing in z'-x' plane. Default = 0.0

NCR = (I) Failure criteria. Any of: 0, ±1, ±2, ±3, ±4; negative values cause a plasticity treatment, ie no stress drop
NCR=0 - Elastic material, no failure
NCR=1 — Maximum stress failuire criteria
NCR=2 — Tsai-Wu failuire criteria
NCR=3 — Hashin failure criteria
NCR=4 — Christensen failure criteria
Default = 0

sigyc1 = (F) Applicable only if NCR > 1. The compressive yield stress(strain) in the x' direction of the material. Default = 0.0

sigyc2 = (F) Applicable only if NCR > 1. The compressive yield stress(strain) in the y' direction of the material. Default = 0.0

tau = (F) Number of timesteps for dropping stresses after failure

Notes: 1. The last 3 slots of the EPLS array contain failure indicators: A value greater than 0 in the last slot indicates in-plane shear failure. A value greater than 0 in the second to last slot indicates failure in the y', or matrix direction, and a value greater than 0 in the 3rd to last slot indicates failure in the x' or fiber direction.

2. None of the models currently consider transverse shear failure.

3. For the maximum stress criterion (NCR=1), fiber (x') and in-plane shear properties are degraded when the stress in the x' direction exceeds yield. In-plane shear and y' properties are degraded if either the shear stress or y' stress exceeds yield.

$$\frac{\sigma_{11}^2}{X^2} \geq 1 \qquad \textit{Fiber} \quad \text{mode}$$

$$\frac{\sigma_{22}^2}{Y^2} \geq 1 \qquad \textit{Matrix} \quad \text{mode}$$

$$\frac{\tau_{12}^2}{S^2} \geq 1 \qquad \textit{Shear} \quad \text{mode}$$

where,

$$X = \begin{cases} X_t & \textit{if} \quad \sigma_{11} > 0 \\ X_c & \textit{if} \quad \sigma_{11} < 0 \end{cases} ; \quad Y = \begin{cases} Y_t & \textit{if} \quad \sigma_{22} > 0 \\ Y_c & \textit{if} \quad \sigma_{22} < 0 \end{cases}$$

4. For the Tsai-Wu criterion(NCR=2), all in-plane properties are degraded if the yield criterion is exceeded.

$$\frac{\sigma_{11}^2}{X_t X_c} + \frac{\sigma_{22}^2}{Y_t Y_c} + \frac{C_{12}\sigma_{11}\sigma_{22}}{\sqrt{X_t X_c Y_t Y_c}} + \frac{\sigma_{11}}{X_t - X_c} + \frac{\sigma_{22}}{Y_t - Y_c} + \frac{\tau_{12}}{S^2} \geq 1$$

5. For the Hashin criterion (NCR=3), fiber (x') and in-plane shear properties are degraded when fiber yield is detected. In-plane shear and y' properties are degraded if matrix yield is detected.

$$\frac{\sigma_{11}^2}{X_t^2} + \frac{\tau_{12}^2}{S^2} \geq 1 \qquad \textit{Tensile} \quad \textit{fiber} \quad \text{mode}, \quad \sigma_{11} > 0$$

$$\frac{\sigma_{11}^2}{X_c^2} \geq 1 \qquad \textit{Compressive} \quad \textit{fiber} \quad \text{mode}, \quad \sigma_{11} < 0$$

$$\frac{\sigma_{22}^2}{Y_t^2} + \frac{\tau_{12}^2}{S^2} \geq 1 \qquad \textit{Tensile} \quad \textit{matrix} \quad \text{mode}, \quad \sigma_{22} > 0$$

$$\frac{\sigma_{22}^2}{4S^2} + \left(\frac{\sigma_{22}}{Y_c}\right)\left(\frac{Y_c^2}{4S^2} - 1\right) + \frac{\tau_{12}^2}{S^2} \geq 1 \quad \textit{Compressive} \quad \textit{matrix} \quad \text{mode}, \quad \sigma_{22} < 0$$

where,

$$X = \begin{cases} X_t & \textit{if} \quad \sigma_{11} > 0 \\ X_c & \textit{if} \quad \sigma_{11} < 0 \end{cases} ; \quad Y = \begin{cases} Y_t & \textit{if} \quad \sigma_{22} > 0 \\ Y_c & \textit{if} \quad \sigma_{22} < 0 \end{cases}$$

6. The Christensen criterion is based on strains rather than stresses, and the inputs are interpreted as strains at yield. Again, fiber (x') and in-plane shear properties are degraded when fiber yeld is detected. In-plane shear and y' properties are degraded if matrix yield is detected.

7. By default, the material x',y',z' directions coincide with the local shell directions. The **ANGL** subcommand can be used to change the material directions.

THEX (thermal expansion) Subcommand
SHEL (shell) Input Group **SHEL-THEX**

Purpose: To define thermal expansion coefficients for a shell material.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must follow the **PROP** subcommand that defines the material.

PCOM SCOM input parameters

THEX *mname alphax alphay alphaz*

mname = (C) Name of the material. No default.

alphax = (F) Thermal expansion constant in the local x'-direction of the material. No default.

alphay = (F) Thermal expansion constant in the local y'-direction of the material. Default=*alphax*.

alphaz = (F) Thermal expansion constant in the local z'-direction of the material. Default=*alphax*.

Note: Use the **ANGL** subcommand to define material directions that do not coincide with the local shell directions..

AVIS (artificial viscosity) Subcommand
SHEL Input Group **SHEL-AVIS**

Purpose: To define artificial viscosity (damping) for a shell material.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must follow the **PROP** subcommand defining material *propname*, except for ALL option.

PCOM	SCOM	input parameters
	AVIS	*propname* MATR *alinear aquadratic cmpopt strnrat*
	or	
	AVIS	*propname* CRIT *frequency fracdmp ellength*

propname = (C) Material property name. Enter ALL to set viscosity for all shell elements (except those that have been set explicitly). No default.

option = (C) Damping specification option. ANY of: MATR or CRIT. Set to MATR if specifying damping in terms of linear and quadratic viscosity coefficients. Set to CRIT if specifying damping loss as a fraction of critical damping.No default.

alinear = (F) Linear viscosity coefficient. Default = 0.06.

aquadratic = (F) Quadratic viscosity coefficient. Default = 1.2.

cmpopt = (F) Compression only or both tension and compression. Either COMP or BOTH. Default = BOTH.

strnrat = (F) Estimate of maximum strain rate. Used to compute stable timestep when aqudratic is nonzero. Default = 0.

frequency = (F) Center frequency (cycles/unit time) at which the viscoelastic damping model matches the damping vs. frequency curve No default.

fracdmp = (F) Fraction of critical damping. No default.

ellength = (F) Minimum length of elements to which this command applies. No default.

Notes:
1. Artificial viscosity is used to reduce numerical noise and to approximate shocks by smearing them out over a few elements.
2. For shells it is applied only to axial strains, not to transverse shear strains.

INFO (information) Subcommand
SHEL (shell) Input Group **SHEL-INFO**

Purpose: To print useful information to aid in debugging models with shells.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **SHEL** Group: Must follow **SHEL** command group that defined the shells.

PCOM SCOM input parameters

INFO *option* [*parameters*]

option = (I) Any of INDX, LAYR or IJK.

For option = INDX, the bounding nodes and asssociated layers are printed for the requested shell element.

INFO INDX *shelindex*

shelindex = (I) Shell element number.

For option = LAYR, the shell element number and bounding nodes are printed for the given layer.

INFO LAYR *layer*

For option = IJK, Prints a list of shell elements connected to the given node.

INFO IJK *i j k*

i,j,k= (I) The (i,j,k) of a node.

LAYR (layer) Subcommand
SHEL (shell) Input Group **SHEL-LAYR**

Purpose: To define a through-the-thickness layering for shell elements.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence
Within **SHEL** Group: Must precede any **ELEM** subcommand that references this layer definition.

PCOM SCOM input parameters

LAYR *(option) lname thick nlayer pname(1) ... pname(nlayer) & hginpln hgnorm hgrot rotfac*

or

LAYR *(option) lname thick numlayer pname hginpln hgnorm hgrot rotfac*

option = (C) For general connectivity models only, set to GCON. Otherwise ignore this parameter. If set to GCON, then the thickness will be scaled by the active gc scale factor.

lname = (C) Name of this shell layer definition.
No default.

thick = (F) Total thickness of all layers in this definition.
No default.

nlayer = (I) Number of layers through the thickness for this definition. No default.

pname(i) = (C) Material property name for layer (i). If all layers use the same material property definition, a single *pname* can be input. No default.

hginpln = (F) Hourglass suppression constant for inplane (membrane) modes
.
Default = 0.01. Active for 3D only.

hgnorm = (F) The hourglass suppression constant for normal to the element plane (bending) modes.
Default = 0.01. Active for 3D only.

hgrot = (F) Hourglass suppression constant for rotational (transverse shear) modes.

Default = 0.01 Active for 3D only.

rotfac = (F) Scale factor for rotary inertia. Scales rotary inertia so that in-plane modes control timestep.

Default = 2.0

ELEM (elements) Subcommand
SHEL (shell) Input Group **SHEL-ELEM**

Purpose: To define the connectivity for a group of shell elements and specify the associated layer definition. Shell elements are typically coincident with continuum element faces.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must follow the **LAYR** subcommand that defines the shell layer properties for the defined elements.

PCOM SCOM input parameters

ELEM *lname ibegin iend jbegin jend kbegin kend (mat1) (mat2)*

lname = (C) Name of the shell layer definition.
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes that bound the group of shell elements defined.
Default: *ibegin* = 1, *iend* = maximum I

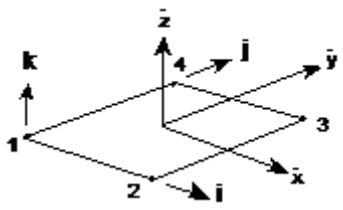
jbegin, jend = (I) Beginning and ending J-indices of nodes that bound the group of shell elements defined.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes that bound the group of shell elements defined (3D only).
Default: *kbegin* = 1, *kend* = maximum K

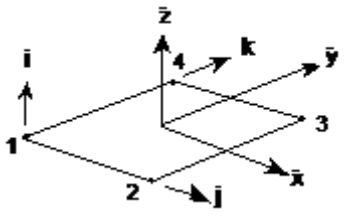
mat1, mat2 = (C) If entered, shells are only generated on the *mat1*/*mat2* interface

---

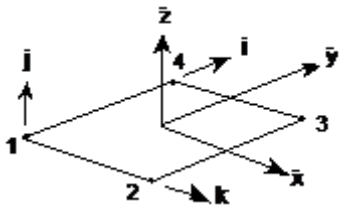
Notes: 1. For 3D models, *ibegin* must equal *iend, jbegin* must equal *jend,* or *kbegin* must equal *kend.* For 2D models, *ibegin* must equal *iend* or *jbegin* must equal *jend.*
2. The four nodes that define each shell element are assumed to be coplanar.
3. The local coordinate systems in which stress and strain quantities are defined are:



I-J face element



J-K face element



K-I face element

SNGL (nodal connectivity) Subcommand
SHEL (shell) Input Group **SHEL-SNGL**

Purpose: To define an arbitrary nodal connectivity for a single shell element and specify the associated layer definition.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must follow the **LAYR** subcommand that defines the shell layer properties for the defined element.

PCOM	SCOM	input parameters
(for 3D)		
	SNGL	*lname i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4*
(for 2D)		
	SNGL	*lname i1 j1 k1 i2 j2 k2*

lname = (C) Name of the shell layer definition. No default.

i1, j1, k1 = (I) I, J, and K indices of node 1 for the shell element. No default.

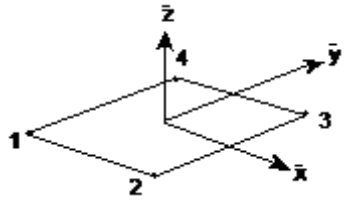
i2, j2, k2 = (I) I, J, and K indices of node 2 for the shell element. No default.

i3, j3, k3 = (I) I, J, and K indices of node 3 for the shell element. No default.

i4, j4, k4 = (I) I, J, and K indices of node 4 for the shell element. No default.

Notes:

1. The four nodes that define the shell element are assumed to be coplanar.
2. If a large deformation analysis is being performed, all nodes specified by the **SNGL** subcommand must be in the same zone.
3. The local coordinate system in which stress and strain quantities are defined is:



Local coordinate system for **SNGL** defined shell elements

GCON (general connectivity) Subcommand
SHEL (shell) Input Group **SHEL-SNGL**

Purpose: To define a shell element using general connectity syntax for node mapping. See Section 2: *Importing General Connectivity Models*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must follow the **LAYR** subcommand that defines the shell layer properties for the defined element.

PCOM	SCOM	input parameters
(for 3D)		
	GCON	*lname nodegc1 nodegc2 nodegc3 nodegc4 (groupnames)*
(for 2D)		
	GCON	*lname nodegc1 nodegc2 (labelname)*

lname = (C) Name of the shell layer definition. No default.

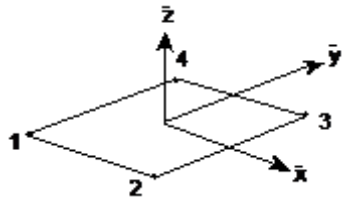
nodegc1,nodegc2, nodegc3, nodegc4 = (I) Local general connectivity node numbers defining the connectivity for this element. Note that triangular elements can be defined by setting *nodegc3*=*nodegc4*
No defaults.

groupnames = (CSet of group names can be appended to the end of this subcommand. Information about elements of a particular group can be accessed using the **SYMB** #GET GCON option.

labelname = (C) A label to assign to this shell element to identify it within other commands.

Notes:

1. The four nodes that define the shell element are assumed to be coplanar.
2. If a large deformation analysis is being performed, all nodes specified by the **GCON** subcommand must be in the same zone.
3. The local coordinate system in which stress and strain quantities are defined is:



Local coordinate system for **GCON** defined shell elements

DEL (delete) Subcommand
SHEL (shell) Input Group **SHEL-DEL**

Purpose: To delete shell elements.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede PRCS.
within **SHEL** Group: Must follow the **ELEM** or **SNGL** subcommands that generated the elements.

PCOM	SCOM	input parameters
	DEL	*layrname ibeg iend jbeg jend kbeg kend*

layrname = (C) *layrname* of shell elements within the deletion window to be deleted. Default: all shell elements within the window are deleted.

ibe.g., iend = (I) Bounding i-indices of the deletion window. Default: *ibeg* = 1, *iend* = max i.

jbe.g., jend = (I) Bounding j-indicies of the deletion window. Default: *jbeg* = 1, *jend* = max j.

kbe.g., kend = (I) Bounding k-indicies of the deletion window. Default: *kbeg* = 1, *kend* = max k.

Notes: 1. All shelllements of type *layrname* contained entirely within the deletion window are deleted.
2. This option provides convenient way of generating cutouts.

THIK (thickness) Subcommand
SHEL (shell) Input Group **SHEL-THIK**

Purpose: To specify unequal layer thicknesses for a shell layer model

Use Is: Optional. If not input, the layer model defaults to equal spaced layers.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: None.

PCOM SCOM input parameters

THIK *lname fraction1 fraction2 ... fractionn*

lname = (C) Name of previously defined layer definition. No default.

fraction1, .., fractionn = (F) Fraction of the total shell thickness assigned to each layer. There must be a value for each layer defined for the *lname* layer definition. The sum of all fraction inputs must be 1.0. No default.

OFFS (offset) Subcommand
SHEL (shell) Input Group **SHEL-OFFS**

Purpose: To specify that the shell be offset from the plane of the nodes to which it is assigned.

Use Is: Optional. If not input, the shell is centered in the plane of the nodes which define it..

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: None.

PCOM	SCOM	input parameters
	OFFS	*lname zoffset*

lname = (C) Name of previously defined layer definition.
No default.

zoffset = (F) Offset of the shell in the shell elements local z direction.
Default = 0.0

ANGL (angle) Subcommand
SHEL (shell) Input Group **SHEL-ANGL**

Purpose: To specify local orientation of orthotopic model for each layer of a layer definition.

Use Is: Optional. If not input, the local x' axis is defined as the direction defined by nodes 1 and 2 of each shell element. The y' axis is the cross product of x' with the z' outward normal vector.

Multiple Use: Allowed.

Order Dependence within **SHEL** Group: Must follow the LAYR command that defined the layering definition.

PCOM SCOM input parameters

ANGL *lname* ANGL *angle1 angle2 ... anglen*
or
ANGL *lname* EULR *eangle1 eangle2 ... eanglen*
or
ANGL *lname* VECT *xvec1 yvec1 zvec1 xvec2 yvec2 zvec2 ... &*
xvecn yvecn zvecn

lname = (C) Name of previously defined layer definition.
No default.

anglen = (F) Angle that the orthotropic x' axis makes with the vector defined by node 1 and 2 for each shell
No default.

eanglen = (F) Euler angle that the orthotropic x' axis makes with the vector defined by node 1 and 2 for each shell
No default.

vectxn, vectyn, vectzn = (F) x, y, and z components of vector defining orthotropic x' axis when projected onto surface of each shell element.
No default.

BUFR (Buffer) Subcommand
SHEL (Shell) Input Group **SHEL-BUFR**

Purpose: Change the incremental memory block size allocated to arrays created with SHEL commands. (NOTE: this command is typically unnecessary unless prohibitively large read-in times are experienced)

Use Is: Optional. If not input, default memory allocation is used.

Multiple Use: Allowed.

Order Dependence
Within **SHEL** Group: Must precede **ELEM** subcommands.

PCOM SCOM input parameters

BUFR *nsiz_elem*

nsiz_elem = (I) The number of elements created with ELEM, SNGL and GCON commands to allocate space for at a time.
Default = 1000. See Note 1.

Notes: 1. This option is not typically required. It should only be used if a model contains a very large number of shell elements and the read-in time is long. The shell arrays are resized after read-in so it is not possible to over-allocate memory, but care should be taken not to enter a number larger than the memory space immediately available on the machine.

LABL (Label) Subcommand
SHEL (Shell) Input Group **SHEL-LABL**

Purpose: To assign character string labels to shell elements for grouping and identifying shell elements within other commands

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SHEL** Group: Must follow the **ELEM** subcommand that defines the shells being labeled.

PCOM SCOM input parameters

LABL *labelname ibegin iend jbegin jend kbegin kend touch_option*

labelname = (C) The label to assign to shell elements within the specified region.
No default.

ibegin, iend = (I) The beginning and ending I-indices of nodes which bound the group of shell elements defined.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) The beginning and ending J-indices of nodes which bound the group of shell elements defined.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) The beginning and ending K-indices of nodes which bound the group of shell elements defined (3-D only).
Default: *kbegin* = 1, *kend* = maximum K

touch_option = (C) Option determines whether an element must be contained entirely within the node range. Either of:
ALL = entire element must be contained
TOUCH = element must only touch node range
Default = TOUCH.

Notes: 1. Shell labels are stored in the SHEL_LABEL(i) array, where i ranges from 1 to the number of shells in the model.

PRNT (print) Subcommand
SHEL (shell) Input Group **SHEL-PRNT**

Purpose: To print shell element information.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **SHEL** Group: None.

PCOM SCOM input parameters

PRNT *option*

option = (C) Print option.
One of:
PROP = print material property models
LAYR = print layer definitions
ELEM = print element connectivity and layer type
ALL = print material property, layer, and element data

Default = ALL

SHOW Input Command

SHOW

Purpose: To display a data group stored in the Data Manager. The data are printed to the Job Output File in a simple, normalized form. Any active, real number, data array may be displayed at any time during an analysis.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the point in the job processing at which the data array has become active in the Data Manager. For most field variables, this is after the **PRCS** command.

PCOM _____ input parameters _____

SHOW *dataname iaxis ibegin iend jbegin jend kbegin kend normvalue*

dataname = (C) Name of data array containing variables to be displayed. For example: XVEL, SGYY, etc.
No default.

iaxis = (C) Indices axis normal to the plane of data to be displayed. Any of I, J or K. Default = K

ibegin, iend = (I) Beginning and ending I-indices for data array locations to be displayed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for data array locations to be displayed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices for data array locations to be displayed.
Default: *kbegin* = 1, *kend* = maximum K

normvalue = (F) Unit normalizing value.
Default: *normvalue* = .1 x (maximum absolute value in data array).

Note: . Normalized values are printed as: ± 1, 2, ..., 9, A, B, ..., Y, Z, *, *, ..., * as the values grow larger (either positive or negative). The value "+6" would be printed for *dataname*(I,J) if: 6 x *normvalue* ≤ *dataname*(I,J) < 7 x *normvalue*

Example: The following command produces a printed map of normalized values for the ε_{XX} component of the strain field for the entire grid. A plane of data is displayed for each K-index:
SHOW EPXX

SITE Input Command

Purpose: To define the site model by assigning material properties to each continuum element of the grid.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence: Must follow the **GEOM** and **MATR** commands.

PCOM SCOM input parameters

SITE

BLOK	*matname option xbegin xend ybegin yend zbegin zend & ibegin iend jbegin jend kbegin kend*
CYLN	*matname (iaxis) cbegin cend center1 center2 radbegin radend & holbegin holend thetabeg thetaend ibegin iend jbegin jend kbegin kend*
ELIP	*matname xcenter ycenter zcenter axisname a b c*
FLIT	*matname axisname Ox Oy Oz a b c shape & ibegin iend jbegin jend kbegin kend*
LOCK	*matnam1 matnam2 matnam3…*
MMAP	*format filename ibeg iend jbeg jend kbeg kend nxbin nybin nzbin & axisname scalc scalx scaly scalz midofs*
POLY	*matname (axis) ibegin iend jbegin jend kbegin kend zbeg zend & x1 y1 x2 y2 x3 y3 … xN yN*
REGN	*matname ibegin iend jbegin jend kbegin kend*
REGNCOPY	*(option) (option) (option) ibegf iendf jbegf jendf kbegf & kendf ibegt iendt jbegt jendt kbegt kendt*
REGNDUPL	*ibegf iendf jbegf jendf (kbegf)(kendf) dupdir ncopy invertopt*
SPHR	*matname xcenter ycenter zcenter radius*
SWAP	*matname matnew ibegin iend jbegin jend kbegin kend*
TETR	*matname axisname x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4 & ibegin iend jbegin jend kbegin kend*

SITE Input Command (cont.)

SITE

PCOM	SCOM	input parameters
	TABL	*option filename scale xshift yshift zshift &* *ibegin iend jbegin jend kbegin kend resize*
	TORUS	*matname (axis) xcenter ycenter zcenter rbig rsmall thetabeg thetaend phibeg phiend* *ibegin iend jbegin jend kbegin kend*
	ARAY	*arrayname matname vmin vmax ibegin iend jbegin jend kbegin kend*
	OBJT	*objectname matname*
	PRNT	*iaxis ibegin iend jbegin jend kbegin kend*
	END	

Notes: 1. All elements of the grid must be assigned a valid material name, including VOID, or processing is halted.
2. If the material for a continuum element is specified more than once, the last specification is used to define the element's material.

BLOK (block region) Subcommand
SITE Input Group **SITE-BLOK**

Purpose: To specify the material properties for a portion of the computational grid that falls within a rectangular region defined in xyz space. Implemented only for 2D and 3D models. The search may be restricted to an ijk window of the grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **SITE** Group: None.

PCOM SCOM input parameters

BLOK *matname option (axisname) xbegin xend ybegin yend zbegin zend & ibegin iend jbegin jend kbegin kend*

matname = (C) Name of the material to assign to the region.
No default.

option = (C) Assignment option. Any of: CENT or PART
If CENT: An element is assigned *matname* properties if its centroid is within the defined block
If PART: An element is assigned *matname* properties if any portion of the element is within the defined block
Default = CENT

axisname = (C) Optional input parameter. If the orientation of the rectangular block being defined does not align with the standard global axes of the model (STND), then input *axisname* as the name of the local coordinate system for this block. This axis definition must have been previously defined by the **AXIS** command group.
Default = STND

xbegin, xend = (F) Beginning and ending x-coordinates that define the limits of the block region.
No default.

ybegin, yend = (F) Beginning and ending y-coordinates that define the limits of the block region.
No default.

zbegin, zend = (F) Beginning and ending z-coordinates that define the limits of the block region.
Input only for 3D models.
No default.

ibegin, iend, = (I) Beginning and ending I-indices of nodes bounding the search region. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the search region. Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the search region. Default: *kbegin*=1, *kend*=maximum K

CYLN (cylindrical region) Subcommand
SITE Input Group **SITE-CYLN**

Purpose: To specify the material properties for a cylindrical region of the computational grid. This command has been generalized to work with fairly general conical sectors. Applies only to a standard partition grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

CYLN *matname (axisname) iaxis cbegin cend center1 center2 radbegin radend holbegin & holend thetabeg thetaend ibegin iend jbegin jend kbegin kend*

matname = (C) Name of the material to assign to the region.
No default.

axisname = (C) Optional input parameter. If the orientation of the cylinder axis does not align with the standard global axes of the model (STND), input *axisname* as the name of the local coordinate system for this cylinder. This axis definition must have been previously defined by the **AXIS** command group.
Default = STND

iaxis = (C) Required if *axisname* is specified, optional if *axisname* is defaulted. *iaxis* defines the longitudinal coordinate axis of the cylinder.
Any of: X, Y or Z. Default = Z

cbegin, cend = (F) Beginning and ending *iaxis* coordinate values along the length of the cylindrical region.
No default.

center1, center2 = (F) Coordinates of the regions center axis.
Default = 0.0
If *iaxis* = X:
center1 = y-coordinate, *center2* = z-coordinate
If *iaxis* = Y:
center1 = z-coordinate, *center2* = x-coordinate
If *iaxis* = Z:
center1 = x-coordinate, *center2* = y-coordinate

radbegin = (F) Outer radius of the cylindrical region at *cbegin*. Default = 0.

radend = (F) Outer radius of the cylindrical region at *cend*. Default = *radbegin*.

holbegin = (F) Inner radius of the cylindrical region at *cbegin* Default = *0.*

holend = (F) Inner radius of the cylindrical region at *cend*. Default = *holbegin*

thetabegin = (F) Beginning angle of the sector [degrees]. Theta is measured from x if iaxis = z, from y if iaxis = x, and from z if iaxis = y. Default =0.

thetaend = (F) Ending angle of the sector [degrees]. Default = 360.

ibegin, iend, = (I) Beginning and ending I-indices of nodes bounding the search region. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the search region. Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the search region. Default: *kbegin*=1, *kend*=maximum K

Note: If the centroid of an element falls within the cylindrical region defined by the **CYLN** subcommand, the element is assigned the material name specified.

ELIP (ellipsoidal region) Subcommand
SITE Input Group **SITE-ELIP**

Purpose: To specify the material properties for an ellipsoidal region of the computational grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

ELIP *matname xcenter ycenter zcenter axisname a b c*

matname = (C) Name of the material to assign to the region. No default.

xcenter, ycenter, zcenter= (F) x,y, and z coordinates of the region's center. Default = 0.

axisname= (C) Name of a local axis system which has been previously defined with an AXIS command. Defines the orientation of the ellipsoid. Default = STND (Global axes).

a = (F) Maximum radius in the local x' direction. No default.

b = (F) Maximum radius in the local y' direction. No default.

c = (F) Maximum radius in the local z' direction. No default.

Note: If the centroid of an element falls within the ellipsoidal region defined by the **ELIP** subcommand, the element is assigned the material name specified.

FLIT (fillet) Subcommand
SITE Input Group **SITE-FLIT**

Purpose: To define a prismatic fillet.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **SITE** Group: None.

PCOM SCOM input parameters

FLIT *matname axisname Ox Oy Oz a b c shape ibegin iend jbegin jend kbegin kend*

matname= (C) Material to be assigned to this fillet region. No default.

axisname= (C) Either the name of a previously defined local axis system (see AXIS command) or STND for global axes. Default = STND.

Ox, Oy, Oz = (F) Origin of the fillet region, Default = (0, 0, 0)

a,b,c= (F) Fillet dimensions. *a* is thecross-sectional dimension in the x' direction, *b* is the cross-sectional dimension in the y' direction, and *c* is the length in the z' direction.

shape= (C) Shape of the fillet cross section, any of: TRI, CVEX, or CCAV. These produce triangular, convex ellipse and concave ellipse respectively. Default = TRI.

ibegin, iend= (I) Beginning and ending I-indicies of nodes bounding the search region. Default *ibegin* =1, *iend* = maximum I..

jbegin, jend= (I) Beginning and ending J-indicies of nodes bounding the search region. Default *jbegin* =1, *jend* = maximum J.

kbegin, kend= (I) Beginning and ending K-indicies of nodes bounding the search region. Default *kbegin* =1, *kend* = maximum K.

LOCK Subcommand
SITE Input Group **SITE-LOCK**

Purpose: To LOCK materials from being overwritten by subsequent REGN, BLOK, SPHR, POLY, TORUS, ELIP, FLIT, TETR, HEX, MMAP, ARAY, OBJT subcommands.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM	SCOM	input parameters
	LOCK	*matnam1 matnam2 matnam3…*

matnam(n) = (C) names of materials to be locked. Multiple lock commands add to the list of locked materials. Enter OFF to clear all locks. No default.

Notes: 1. By default, materials are overwritten by subsequent subcommands. LOCK provides a mechanism to prevent this.

MMAP (Material Map) Subcommand
SITE Input Group **SITE-MMAP**

Purpose: Maps the material IDs from a general tetrahedral mesh generated by a 3rd party preprocessor to a standard partition grid of hexahedral elements in FLEX.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence
Within **SITE** Group: None. Must follow AXIS primary command.

PCOM SCOM input parameters

MMAP *format filename ibeg iend jbeg jend kbeg kend nxbin nybin nzbin & axisname scalc scalx scaly scalz midofs*

format = (C) The name of the format of the input ASCII file. Supported formats are NASTRAN bulk data files and ExodusII (from CUBIT).
No default.

filename = (C) The name of the input ASCII file.
No default.

ibeg, iend = (I) Beginning and ending I-indices of nodes bounding the mapped region. Default: *ibeg* = 1, *iend* = maximum I

jbeg, jend = (I) Beginning and ending J-indices of nodes bounding the mapped region. Default: *jbeg* =1, *jend*=maximum J

kbeg, kend = (I) Beginning and ending K-indices of nodes bounding the mapped region. Default: *kbeg*=1, *kend*=maximum K

nxbin, nybin, nzbin = (I) The number of subsearching bins in the x, y, and z directions. Default: *nxbin=(iend-ibeg)/2, nybin=(jend-jbeg)/2, nzbin=(kend-kbeg)/2*

axisname = (C) The name of the previously defined coordinate transformation.
Default is no transformation.

scalc = (F) Scale factor to transform map coordinates to global FLEX coordinates. Default = 1.0

(The **MMAP** subcommand is continued on the next page)

scalx = (F) X-direction distortion scale factor to transform map coordinates to global FLEX coordinates. Default = 1.0

scaly = (F) Y-direction distortion scale factor to transform map coordinates to global FLEX coordinates. Default = 1.0

scalz = (F) Z-direction distortion scale factor to transform map coordinates to global FLEX coordinates. Default = 1.0

midofs = (I) Material ID number offset. The ID number from the tetrahedral model is shifted by *midofs* before mapping to the FLEX grid. Can be positive or negative. Default = 0

Notes:
1. The entire "variables:" block should be deleted from the exodusII ASCII file.
2. The time to perform the material mapping procedure is considerably reduced by increasing the number of bins. The number of bins should not exceed the number of tetrahedrons. The defaults should be optimal
3. The exodus II file contains material id numbers that correspond to the material id numbers in FLEX that are assigned to materials in the order input.

POLY (Polygonal prism) Subcommand
SITE Input Group **SITE-POLY**

Purpose: To specify the material properties for a polygonal prism region of the computational grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM SCOM input parameters

POLY *matname (axis) ibegin iend jbegin jend kbegin kend zbeg zend &*
x1 y1 x2 y2 x3 y3 … xN yN

matname = (C) The name of the material to assign to the region. No default.

axis= (C) The local Cartesian axis system. Must be previously defined by an AXIS primary command. Default = stnd (global coordinates)

ibegin, iend, = (I) Beginning and ending I-indices of nodes bounding the search region. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the search region. Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the search region. Default: *kbegin*=1, *kend*=maximum K

zbeg, zend = (F) The beginning and ending local z-coordinates to define a cylinder in 3D.
Set to default for 2D models. No default for 3D.

x1 y1 x2 y2 x3 y3 … xN yN = (F) Pairs of coordinates defining the vertices of the polygonal shape in the local coordinate system. No default.

Notes: 1. If the centroid of an element falls within the polygonal region defined by the **POLY** subcommand, the element is assigned the material name specified.
2. A minimum of 3 vertices must be specified and a maximum of 50 vertices may be specified.
3. The vertices must be entered in order either clockwise or counter-clockwise around the shape.
4. The shape must be convex, that is having no reentrant corners.
5. In 3D, the polygonal shape is extruded along the local z-axis to the bounds specified by *zbeg* and *zend*.

REGN (region) Subcommand
SITE Input Group **SITE-REGN**

Purpose: To specify the material properties for a rectangular region of the computational grid in IJK space.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

REGN *matname ibegin iend jbegin jend kbegin kend*

matname = (C) Name of material to assign to the region.
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the region.
Default: *kbegin* = 1, *kend* = maximum K

REGNCOPY Subcommand
SITE Input Group **SITE-REGNCOPY**

Purpose: To change a material over a rectangular region of the computational grid in IJK space.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM SCOM input parameters

REGNCOPY *(option) (option) (option) ibegf iendf jbegf jendf kbegf & kendf ibegt iendt jbegt jendt kbegt kendt*

option = (C) Enter any of revi, revj, revk to reflect in that direction.
Default = no reflection.

Ibegf…, kendf = (I) Bounding nodes of region to be copied
Default: Entire grid.

ibegt,…kendt = (I) Bounding nodes of region to be filled.
Default: Entire grid.

REGNDUPL Subcommand
SITE Input Group **SITE-REGNDUPL**

Purpose: To duplicate the materials in a given rectangular region of the computational grid in IJK space. Multiple copies for a periodic structure are easily accommodated.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM SCOM input parameters

REGNDUPL *ibegf iendf jbegf jendf (kbegf)(kendf) dupdir ncopy invertopt*

ibegf…, kendf = (I) Bounding nodes of region to be copied
Do not enter *kbegf, kendf* for 2D models.
Default: Entire grid.

dupdir = (I) Duplication direction. Any of i,j,k.
Default: i

ncopy = (I) Number of times to duplicate.
Default: 1

invertopt = (I) Enter INVERT to invert each copy relative to the previous one.
Default: No inversion.

SPHR (spherical region) Subcommand
SITE Input Group **SITE-SPHR**

Purpose: To specify the material properties for a spherical region of the computational grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

SPHR *matname xcenter ycenter zcenter radius radin dum1 dum2 dum3 dum4 ibegin iend jbegin jend kbegin kend*

matname = (C) Name of the material to assign to the region. No default.

xcenter, ycenter, zcenter = (F) Coordinates of the regions center Default = 0.0

radius = (F) Radius of the spherical region. No default.

radin = (F) Inner radius of region. Default = 0.

Dum1,...dum4 = (F) Angular controls. Not currently active.

ibegin, iend, = (I) Beginning and ending I-indices of nodes bounding the search region. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the search region. Default: *jbegin*=1, *jend*=maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the search region. Default: *kbegin*=1, *kend*=maximum K

Note: If the centroid of an element falls within the spherical region defined by the **SPHR** subcommand, the element is assigned the material name specified.

SWAP Subcommand
SITE Input Group

SITE-SWAP

Purpose: To change a material over a rectangular region of the computational grid in IJK space.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM SCOM input parameters

SWAP *matname matnew ibegin iend jbegin jend kbegin kend*

matname = (C) The name of the material to be changed.
No default.

matnew = (C) The new name of the material. Must have been defined under MATR
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the region.
Default: *kbegin* = 1, *kend* = maximum K

TETR(Tetrahedron) Subcommand
SITE Input Group **SITE-TETR**

Purpose: To assign a materialproperties to elements within a tetrahedron shaped volume within the mesh.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM SCOM input parameters

TETR *matname axisname x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4 &*
ibegin iend jbegin jend kbegin kend

matname= (C) The material to be assigned to this tetrahedral region. No default.

axisname= (C) Either the name of a previously defined local axis system (see AXIS command) or STND for global axes. Default = STND.

x1,y1,...z4 = (F) The coordinates of the tetrahedron's verticies in local coordinates. Default = (0, 0, 0)

ibegin, iend= (I) Beginning and ending I-indicies of nodes bounding the search region. Default *ibegin* =1, *iend* = maximum I..

jbegin, jend= (I) Beginning and ending J-indicies of nodes bounding the search region. Default *jbegin* =1, *jend* = maximum J.

kbegin, kend= (I) Beginning and ending K-indicies of nodes bounding the search region. Default *kbegin* =1, *kend* = maximum K.

TORUS (Toroidal region) Subcommand
SITE Input Group

Purpose: To specify the material properties for a toroidal region of the computational grid.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **SITE** Group: None.

PCOM SCOM _____ input parameters _____

TORUS *matname (axis) xcenter ycenter zcenter rbig rsmall thetabeg thetaend phibeg phiend ibegin iend jbegin jend kbegin kend*

matname = (C) The name of the material to assign to the region. No default.

axis= (C) The local Cartesian axis system. Must be opreviously defined by an AXIS primary command. Default = stnd (global coordinates)

xcenter, ycenter, zcenter = (F) The coordinates of the region's center Default = 0.0

rbig = (F) The large radius of the torus (in x', y' plane). No default.

rsmall = (F) The small radius of the torus. No default.

thetabeg, thetaend = (F) Angular controls on theta (in x', y' plane measured from x') is between -180 and 180. Default = (-180, 180).

phibeg, phiend = (F) Angular controls phi (in rbig, z' plane; measured from + rbig) is between -180, 180. Default = -180, 180.

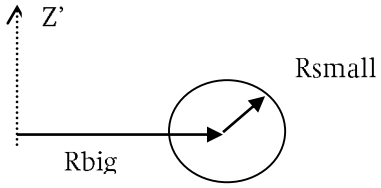
ibegin, iend, = (I) Beginning and ending I-indices of nodes bounding the search region. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the search region. Default: *jbegin*=1, *jend*=maximum J

(continued next page)

PCOM SCOM input parameters

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the search region. Default: *kbegin*=1, *kend*=maximum K



Notes: 1. If the centroid of an element falls within the region defined by the **TORUS** subcommand, the element is assigned the material name specified.

TABL (table) Subcommand
SITE Input Group **SITE-TABL**

Purpose: Assign material properties to grid elements based on an element's coordinates by searching a table that relates material properties to grid coordinates. The table is provided by a separate file. This file may be in ASCII or binary format. This option is used when the variation of materials throughout the model is somewhat random and not easily represented by the standard subcommands used to assign materials to contiguous regions of the grid. The table may be prepared based on experimental data or on some statistical algorithm used to represent the spatial variation of material properties.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

TABL *option filename scale xshift yshift zshift ibegin iend jbegin jend kbegin kend resize*

or

TABL IN *filename axisname xscal yscal zscal ibegin iend jbegin jend kbegin kend resiz (FORE mat1 mat2 ... matn)*

option = (C) set to: IN, or OUT
Default = IN.

filename = (C) The name of the disk file containing the material table. No default.

scale = (F) Scale factor to convert table coordinates into FLEX global coordinate units.
Default = 1.

xshift, yshift zshift = (F) Shift distances in the x-, y- and z-directions to align the table origin with the FLEX global coordinate origin. This coordinate shift is applied after the *scale* factor is applied. See note 3.
Default = 0.

ibegin, iend = (I) Beginning and ending I-indices of nodes which bound the elements of the grid which will query the table for their material properties .
Default: *ibegin* = 1, *iend* = maximum I

(description of **TABL** subcommand continues on next page)

jbegin, jend = (I) Beginning and ending J-indices of nodes which bound the elements of the grid which will query the table for their material properties .
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes which bound the elements of the grid which will query the table for their material properties .
Default: *kbegin* = 1, *kend* = maximum K

resize = (C) If set to RESIZE, the code will redimension the mesh using stored symbol value keypoints. See note 4.
Default = no resizing.

axisname = (C) name of a previsously defined local axis. Enter STND for global coordinates. For this option, coordinate shifts may be specified in the AXIS definition.

xscal, yscal,zscal = (C) scale factors to convert TABL coordinates to Flex coordinates.
Default *xscal* = 1., *yscal=xscal, zscal= xscal.*

FORE = (C) keyword. If enetered, following materials will be foreground — not ovwrwritten by the TABL materials

---

Notes: 1. Only elements which fall within the ijk window defined by ibeg, iend, etc. will be assigned material properties base on table information.

2. If an element within the ijk window has an element centered coordinate which falls outside the boundaries of the table, it will not be assigned a material by the **TABL** subcommand.

3. The coordinates which are contained in the table file, xt,yt and zt are converted into the FLEX global coordinate system, xg, yg, zg by:
   xg = *xshift* + *scale* * xt
   yg = *yshift* + *scale* * yt
   zg = *zshift* + *scale* * zt

4. Symbol values defining the keypoint coordinates must be previously defined in the input file. In the x-direction the symbol names are $x1, $x2, ..., $xN and similarly for y- and z-directions. Keypoints and divisions must also be properly defined in the material table file. *resize* is only applicable for *option* = IN.

ARAY (array) Subcommand
SITE Input Group **SITE-ARAY**

Purpose: To specify the material properties for individual elements in a region of the computational grid based on the value of the elements in a given array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

ARAY *arrayname matname vmin vmax ibegin iend jbegin jend kbegin kend*

arrayname = (C) Name of the data manager array containing values used to determine material type. The array must be dimensioned the same size as the number of elements in the model.
No default.

matname = (C) Name of the material to assign to the elements.
No default.

vmin = (F) Minimum bound of array value.
Default = 0.

vmax = (F) Maximum bound of array value.
Default = 0.

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the region.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the region.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the region.
Default: *kbegin* = 1, *kend* = maximum K

Note: Material type *matname* is assigned to an element if the element value in *arrayname* is greater than or equal to *vmin* and less than or equal to *vmax*.

OBJT (object) Subcommand
SITE Input Group

SITE-OBJT

Purpose: Assigns material to objects defined with the OBJT command.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM SCOM input parameters

OBJT *objectname matname*

objectname = (C) Name of the previously defined object definition. No default.

matname = (C) Name of the previously defined material. No default.

PRNT (print) Subcommand
SITE Input Group **SITE-PRNT**

Purpose: To print a material map for a rectangular region of the computational grid in IJK space.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **SITE** Group: None.

PCOM	SCOM	input parameters
	PRNT	*iaxis ibegin iend jbegin jend kbegin kend*

iaxis = (C) Material map is displayed for each 2D plane proceeding in the *iaxis* direction.
May be any of: I, J or K
Default = K

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the region to be printed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the region to be printed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the region to be printed.
Default: *kbegin* = 1, *kend* = maximum K

STOP Input Command

STOP

Purpose: To terminate the processing of a job in an orderly manner.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM	input parameters
STOP	-none-

Notes:
1. If a **STOP** command is encountered in the middle of the input stream, no input commands following the **STOP** command are processed.
2. If no **STOP** command is found and the end-of-file for the Job Input File is encountered, a **STOP** command is automatically generated, resulting in a normal termination of job processing.
3. A restart file is always written to the Restart Output File upon encountering a **STOP** command. All time histories requested by the **POUT HIST** command are also displayed (unless suppressed) and written to the Time History Output File.

TERM (terminal) Input Command

Purpose: To allow the user to control the routing of standard input and output files to the terminal or to the Job Input and Job Output Files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: None.

PCOM _____ input parameters _____

TERM *inoption outoption*

or

T *inoption outoption*

inoption = (C) Option controlling standard input. Either ON or OFF.
If ON, the user must provide input commands from the terminal keyboard in interactive mode.
If OFF, input commands are read from the Job Input File.
Default: toggle from current state. If currently ON, default is OFF. If currently OFF, default is ON.

outoption = (C) Option controlling standard output . Either ON or OFF.
If ON, standard output from the code is written to the user's terminal screen.
If OFF, standard output is written to the Job Output File.
Default = *inoption*.

Notes:
1. Standard output displayed on a user's terminal is not recorded in the Job Output File.
2. If a user switches the standard input from the Job Input File to the terminal keyboard and then later switches input back to the Job Input File, processing of commands from the Job Input File continues where it left off.

Examples:

```
TERM  /* toggle ON to OFF  or  OFF to ON

TERM OFF  ON   /* input is read from Job Input File, output is displayed at terminal

TERM ON  OFF   /* input from keyboard, output to Job Output File
```

TIME Input Command

TIME

Purpose: To allow the user to control the model time step and initial time.

Use Is: Optional.

Multiple Use: Allowed. Multiple use is provided to allow the user to change the values of *timefactor* and *ratiomin* after the calculation has started for large deformation runs and to reset the start time after the **PRCS** command but before any **EXEC** commands have been input. The *timestep* input parameter is ignored after the first usage of this command. The *timemin* input parameter cannot be entered after the **PRCS** command.

Order Dependence: Except for the exceptions mentioned above, this command must precede the **PRCS** command.

PCOM _____ input parameters _____

TIME *timestep timebegin timefactor ratiomin timemin maxopt*

timestep = (F) Model time step.
Default is internally computed to be largest allowable time step consistent with the Courant stability criteria.

timebegin = (F) Problem time at beginning of execution.
Default = 0.0

timefactor = (F) Time step safety factor for stability.
$timestep \leq timefactor * \Delta t^{max}$
Default = 0.80

ratiomin = (F) For large deformation runs only. *ratiomin* is the minimum allowable value of the ratio of current model time step to baseline model time step. The computation will terminate if the grid distortions during the analysis cause the model time step to fall below this threshold. The baseline model time step is the timestep at the time the **TIME** command is input.
Default = 0.0 which means no minimum timestep ratio.

timemin = (F) Minimum allowable timestep. The density of continuum, shell, beam and/or bar elements with a timestep less than *timemin* is increased to achieve a model timestep equal to *timemin*. See note 3.
Default = 0.0

maxopt= (C) Enter MAX to activate this option. This allows the model timestep to increase (once) the next time it is computed. Typically used after eroding elements that were controlling the model timestep. Theoretically, this can cause instability, so it should be used sparingly and only after a restart has been written.

Notes: 1. The theoretical maximum stable timestep for an element is $\Delta t^{max} = dx/c$, where dx is the critical minimum distance across the element and c is the fastest wave speed of the material assigned to the element. The most critical element in the model determines the model time step.

2. The current model timestep is saved in the TIMSTP array.
3. When setting *timemin*, default *timestep* or set it less than or equal to *timefactor*timemin*.

TITL (title) Input Command

TITL

Purpose: To provide titling information and identifiers for output results. If not input, titling information defaults to blank.

Use Is: Optional but recommended.

Multiple Use: Not allowed.

Order Dependence: None. Recommended to be first command.

PCOM input parameters

TITL <--*id*--> <----------*title*---------->

id = (C) Up to 8 character job identifier used to tag output results associated with the job. If internal blanks are desired, must be enclosed in single quotes. No default.

title = (C) 60-character job title. This input character string need not be enclosed in single quotes even if it includes blanks. All characters following the *id* field are assumed to be part of *title*.
No default.

USER Input Command

USER

Purpose: To provide the user with a command entry point that accesses user-written subroutines. The user must provide the subroutine USRINP and link it with **PZFlex** for this option to be effective. Users should be familiar with the **SOFSHEL** architecture used in **PZFlex** before attempting to write their own subroutines for **PZFlex**. For user-specific pressure and velocity boundary conditions, the user-supplied routines USRPRS and USRVBC provide a guided procedure for satisfying typical users requirements.

Use Is: Optional.

Multiple Use: Depends on the user-written routine.

Order Dependence: Depends on the user-written routine.

PCOM	input parameters
USER	-parameters are dependent on the user written routine-

WNDO (window) Input Command

Purpose: To define a computational window for the model. A computational window is a rectangular region of the model in IJK space that contains the only elements in the model that are computed for the current time step. The computational window expands one grid element in all directions each time step. This option provides a way to conserve computer resources when solving very large problems in which the initial loading affects only a small region of the model.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow the **GRID** command and precede the **PRCS** command.

PCOM _____ input parameters _____

WNDO *ibegin iend jbegin jend kbegin kend*

ibegin, iend = (I) Beginning and ending I-indices of nodes defining the window.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes defining the window.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes defining the window.
Default: *kbegin* = 1, *kend* = maximum K

Notes: 1. If the **WNDO** command is used, any computational zone that does not intersect the current computational window is skipped. Any zone that does intersect the window is truncated to include only those elements that fall within the window. This saves the computational time that would have been spent computing elements thathave no loading applied to them. This savings decreases as the computational window expands to include the entire grid.

2. For piezoelectric models, the computational window must include the electric window. If the user specifies a window that does not encompass the electric window, the specified window is automatically enlarged to bound it.

XFIL (X-files) Input Command

Purpose: To provide for export of portions of a PZFlex model in alternative formats. It is used primarily to export a model in PZFlex's own GCON (general connectivity) format.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: Must follow the **PRCS** command.

PCOM SCOM input parameters

XFIL

FORM	*codeoption*
FILE	*filename*
WRIT	*writeoption tag masscheck*
SYMM	*axis_normal axis_coord fuzz_factor*
END	

FORM Subcommand
XFIL (X-files) Input Group **XFIL-FORM**

Purpose: To specify the form (foreign code name) of the files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **XFIL** Group: Must precede any **WRIT** subcommands.

PCOM SCOM input parameters

FORM *formoption*

formoption = (C) Specify the form of the output file to create.
Any of: GCON =write a file containing PZFlex model information in PZFlex general connectivity format. See Section 2: *Exporting General Connectivity Model*
ANSY = writes a file containing PZFlex continuum element data in ANSYS format
RAD = generates surface data for RADIANCE
MAZ = writes a file containing PZFlex continuum element data for MAZ
DYNA = writes a file containing PZFlex continuum element data in LS-DYNA format

No default.

FILE Subcommand
XFIL (X-files) Input Group **XFIL-FILE**

Purpose: To specify the name of the output file that will be generated using this command group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **XFIL** Group: Must precede the **WRIT** subcommand that writes the file.

PCOM	SCOM	input parameters
	FILE	*filename*

filename = (C) Output file name.
No default.

WRIT (write) Subcommand
XFIL (X-files) Input Group **XFIL-WRIT**

Purpose: To write out a PZFlex model in a format that can be read by a foreign code.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **XFIL** Group: Must follow any **FORM**, **FILE**, and **SYMM** subcommands that apply to this **WRIT** subcommand.

PCOM SCOM input parameters

WRIT *writeoption* (*tag*) *masscheck*

writeoption = (C) Option defining what portion of the model to write.
Any of:
MODL - write out the entire model
No default.

tag = (C) If *formoption* was set to GCON using the **FORM** subcommand, *tag* .is appended as a group label to all nodes and elements exported to the output file.
Default=blank.

masscheck = (C) If set to OFF the code does not check to make sure that nodal masses are connected to structural elements. This allows single-node masses to be output.
Default=ON.

SYMM (symmetry plane) Subcommand
XFIL (X-files) Input Group **XFIL-SYMM**

Purpose: To mirror a model about its symmetry plane when it is exported in GCON format using the **WRIT** subcommand. Mirrored nodes are numbered starting from the last node number of the actual model.

Use Is: Optional.

Multiple Use: Allowed. If input more than once, the last occurrence applies.

Order Dependence within **XFIL** Group: Must precede the **WRIT** subcommand.

PCOM	SCOM	input parameters
	SYMM	*axis_normal axis_coord fuzz_factor*

axis_normal = (C) Global axis direction normal to the plane of symmetry.
Any of: X, Y or Z.
No default.

axis_coord = (F) Value of the *axis_normal* coordinate that lies on the plane of symmetry
Default=0.0.

fuzz_factor = (F) When determining if a node is located on the symmetry plane, any node whose distance from the symmetery plane is less than or equal to *fuzz_factor* is assumed to be a symmetry plane node and is not be mirrored.
Default=0.0001.

ZONE Input Command

Purpose: To define a computational zone. A computational zone is a rectangular region in IJK space.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow **GRID** and precede **PRCS** commands.

PCOM input parameters

ZONE *iratio ibegin iend jbegin jend kbegin kend zonestep*
or
ZONE AUTO

iratio = (I) The ratio of the zone time step to the model time step. Default is internally computed to be the largest *iratio* consistent with stability. In general, it is recommended that this parameter be default. The time step for the zone will be equal to *iratio* x model time step.

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the zone. Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the zone. Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the zone. Default: *kbegin* = 1, *kend* = maximum K

zonestep = (F) Time step override for this zone. Only used if it is desired to run a model with a zone time step which exceeds the stability criteria. It is recommended that this parameter not be used. Default: no override of the *iratio* parameter

AUTO = (C) The keyword auto triggers automatic zoning logic which will attempt to choose some reasonable zones. AUTO cannot be combined with explicit zoning. It is unlikely to be as optimal as zones chosen by hand.

Notes:
1. If no **ZONE** command is input, the grid is zoned automatically in order to satisfy code requirements for zones. No effort is made to optimize the grid zoning to reduce overall run time.
2. If any zones are defined, the user must define zones that include all elements of the grid that are not void. If not, the job will terminate processing of the model.
3. If a region of the grid is all VOID, efficiencies are gained by placing this region in a zone by itself or by not including the region within a zone.
4. Zones must not overlap each other and may not overlap skewed partition boundaries or standard partition mesh boundaries

3 REVIEW - A POSTPROCESSOR

This section briefly describes **Review**, a program for postprocessing and displaying information generated by **PZFlex**. This includes plotting time histories, playing back movies, manipulating PostScript output files, etc. The user interface for **Review** is similar to that of **PZFlex** (i.e., free format input, the use of the primary command/subcommand input structure, the ability to use SYMBOL commands anywhere in the Job Input File, etc). The various commands are described below.

COMMAND STRUCTURE

The command input structure and command syntax are similar to those of **PZFlex** as described at the beginning of Section 3.

COMMAND DEFINITIONS

The command definitions use bold capital letters for all primary commands and subcommands. Input arguments are shown in lower case italic letters. Data types of input arguments are indicated as: (F) = floating point real number, (I) = integer number, (C) = character string, and (X) = a generalized construct that may take different forms and be composed of several arguments and data types. Actual character string inputs are shown with all capital letters.

NOTE:	**All character string inputs to the program, including all commands and input parameters, should be lower case. Only Symbol variable names and values may be upper case.**

A compact list of all commands and arguments is presented first, followed by a detailed description of each command defining its purpose, whether it is required for a typical analysis, if it can be used more than once in the input stream, and whether or not there are any order constraints on its use. Primary commands are presented in alphabetical order. Subcommands follow the primary commands in the order listed for the command group.

Input parameters shown in parentheses are optional and need not be placed on the command line. If an optional parameter is not input, an * (default indicator) is not needed to mark its position on the command line.

Input parameters shown in brackets,[ ], are generalized constructs that may take more than one form and may be composed of more than one input parameter. The command description for the construct should be closely examined in order to understand the possible forms the construct may take.

LIST OF **REVIEW** COMMANDS

PCOM	SCOM	input parameters
		Define local coordinate system axes
ASOURCE		
	FLUID	*density wavespeed*
	FILE	*filename*
	REF	*x-coord y-coord z-coord direction*
	RESOLUTION	*freqmax nstcyc delx*
	POLY	*axisname drivename scale timeshift x1 y1 x2 y2 ... xN yN*
	CIRC	*axisname xcent ycent zcent radO radI drivename scale timeshift*
	RECT	*axisname xmin xmax ymin ymax drivename scale timeshift*
	RUN	*pulse_length*
		Define local coordinate system axes
AXIS		
	DEFN	*axisname axistype x0 y0 z0 x1 y1 z1 x2 y2 z2*
	FORM	*option*
		Convert file formats
CONVERT		
	POV	*format filename_in filename_out*

PCOM SCOM input parameters

Perform input, output and other operations with Data Manager arrays

DATA

ABS	*datanameabs dataname1 dataname2 . . . datanamen*
ADD	*datanamefrom scale datanameto ibegin iend jbegin jend kbegin kend*
CDDO	*dataname filename ibegin iend jbegin jend kbegin kend*
CLOS	*dataname*
COPY	*datanamefrom datanameto*
CPYG	*datanamefrom ibgfrom iincfrom jbgfrom jincfrom kbgfrom kincfrom & datanameto ibgto iendto jbgto jendto kbgto kendto*
DIVIDE	*dataname_denom datamane_numer*
EXPONENT	*dataname exponent*
FFT	*realarray imagarray direction inverse*
FFTSHFT	*array1 (array2 array3 …arrayN) direction*
FILE	*option filename*
HIST	*dataname nvalues filename scaletime scaledata*
IMPORT	THRM *filename matname*
IN	*dataname*
INTERP	*array arraynew*
LOG	*dataname logdataname maxvalue*
MATH	*arraynew* = [algebraic expression involving arrays]
MGR	DIM *dataname ndimension ndim1 (ndim2) (ndim3)*
MGR	MAP *dataname itype mapname*
MIRR	*array arraynew axis option ibeg iend jbeg jend kbeg kend*
MULTIPLY	*dataname1 dataname2*
OPEN	*dataname ndimension ndimi ndimj ndimk datatype*
OUT	*dataname*
OUT0	*dataname*
OUT1	*dataname*
PHAD	*ampfrom phasfrom scale ampto phasto ibegin iend jbegin jend kbegin kend*
POLR	*real imag option amp phas ibegin iend jbegin jend kbegin kend*
SCAL	*dataname scalevalue ibegin iend jbegin jend kbegin kend*

PCOM SCOM input parameters

Compute derivative of time history information

DERV	*filename (label1 label2 . . . labeln)*

Compute time histories using Kirchhoff extrapolation

EXTR	
DATA	*pathname1 pathname2 ....pathnamen*
FILE	*filename (option) fileid*
FREQ	*frequency (attenuation)*
MIRR	*side_normal symm_type* (*side_normal symm_type*) & (*side_normal symm_type*)
SHOW	*display_quantity*
SLVR	*slvrtype*
SURF	*SPHR axisname r begθ endθ nθ begϕ endϕ nϕ*
SURF	*CYLN axisname r zbeg zend nz begθ endθ nθ*
SURF	*QUAD axisname ni nj x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4*
SPOT	*x y z*
TYPE	*option*
CALC	*quad_order stor_opt nsection length iprint*

Bandpass filter time history records

FILT	*filename lowfilter highfilter*

Compute Fourier spectra by taking FFT of time history

FREQ	
FILE	*fftfilename fileid*
TTL	
<----------60 character title for *makefilename* ---------->	
TIME	FROM *filename*
TIME	*ntime timebegin timeinc*
FFT	*filename record*
NFFT	[*record1*] [*record2*]
SFRQ	*freqbegin freqend nfreq*
SPEC	*filename record dampratio*
TYPE	*option*
IMPD	[*curvevolt*] [*curvecharge*] *scalevolt scalecharge*
ADMT	[*curvevolt*] [*curvecharge*] *scalevolt scalecharge*
WNDO	*type option fraction windobegin windoend*
ZIMP	[*curvepres*] [*curvevel*] *density wavespeed*

PCOM	SCOM	input parameters

Graphics display of x-y plots

GRPH		
	ARAT	*aspectratio*
	CLOS	
	LIST	*filename (label) (indxlabel index) (nbegin nend ninc)*
	NVEW	*nview*
	PLOT	*[curve 1] [curve 2] . . . [curve 6]*
	PS	
	SET/PSET	*option parameters*
	SIZE	*xsize ysize xbegin xend ybegin yend nxdiv nydiv*
	TTL	*ititle* (FROM *localfile*)
<----------200 character title information for line = *ititle*---------->		
	VIEW	*iview*
	WRIT	(*options*) *filename*

Manipulate time history files

HIST		
	COMB	*in1filename in2filename outfilename*
	FFLD	*[histname] direction coord_side delcoord wavespeed density* & *timebegin timeend timeinc timeshift typehist outfile*
	SMI	(complex command structure, see command documentation)

Change identifier for file

ID	*filename id*

Integrate time history records

INTG	*filename (label1 label2 . . . labeln)*

List time history records in a file

LIST	*filename (label) (indxlabel index) (nbegin nend ninc)*

PCOM SCOM input parameters

Make new data records (time or frequency domain) by algebraic manipulating of previous records

MAKE

FILE	*makefilename (option) fileid*
TTL	
<---------- 60 character title for *makefilename* ---------->	
TIME/FREQ	FROM *filename*
TIME/FREQ	*npoints valuebegin valueinc*
CURV	< mathematical expression defining new curve >
CURV_ADD	< mathematical expression defining curve to add to previous record >
DFT	*filesignal irecsignal frequency option*
DHML	*fileforc irecforc filestep irecstep*
ENVL	*filesignal irecsignal*
IFFT	*filesignal irecamp irecphas rectype*
LABL	*label i-index j-index k-index x-coord y-coord z-coord*
SHIFT	*filesignal irecsignal timeshift valueshift*
TFRM	*axisname component option filename i-index j-index k-index*
UNWRAP	*filename record shifttype (ntstrt* or *rtstrt)*
WNDO	*type option fraction windobegin windoend*

Allocate RAM memory available to the job

MEM *memory_numeric memory_character option*

Manipulate PostScript files

PS

NCPY	*psfilename ncopies ipagenumber*
SPLT	*psfilename*
CNVR	*psfilename*

Read in time history data

READ *filename pathname nbegin1 nend1 ninc1, . . . , nbeginn nendn nincn*

Scale time history information

SCAL *filename option scalvalue (label1 label2 . . . labeln)*

Shift time history information

SHFT *filename option shift (label1 label2 . . . labeln)*

Stop processing input commands

STOP

Control routing of standard input or output to the terminal

TERM *inoption outoption*

Write out time history records

WRIT *filename (option) pathname (nbegin1 nend1 ninc1, ... , nbeginn nendn nincn)*

ASOURCE (Acoustic Source) Input Command

ASOURCE

Purpose: To generate approximate acoustic sources from simple geometries such that pressure can be extrapolated to target points surrounding the source. This command creates a REVEXT data file containing pressure and pressure gradient on specified surfaces. The REVEXT data file has the same format as a FLXEXT data file and can be used as the input file within an **EXTR** command.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must precede the **EXTR** command that extrapolates pressures from this source.

PCOM SCOM input parameters

ASOURCE

FLUID	*density wavespeed*
FILE	*filename*
REF	*x-coord y-coord z-coord direction*
RESOLUTION	*freqmax nstcyc delx*
POLY	*axisname drivename scale timeshift x1 y1 x2 y2 ... xN yN*
CIRC	*axisname xcent ycent zcent radO radI drivename scale timeshift*
RECT	*axisname xmin xmax ymin ymax drivename scale timeshift*
RUN	*pulse_length*
END	

FLUID Subcommand
ASOURCE (Acoustic Source) Input Group **ASOURCE-FLUID**

Purpose: To set homogeneous fluid parameters surrounding acoustic source.

Use Is: Required.

Multiple Use: Allowed. Repeating the subcommand overwrites previous fluid parameters.

Order Dependence
Within **ASOURCE**: Must precede any **RESOLUTION** and **RUN** subcommands.

PCOM	SCOM	input parameters
	FLUID	*density wavespeed*

density = (F) Fluid density. No Default.

wavespeed = (F) Speed of sound in fluid. No Default.

FILE Subcommand
ASOURCE (Acoustic Source) Input Group **ASOURCE-FILE**

Purpose: To specify the name of the REVEXT file created by this command.

Use Is: Required.

Multiple Use: Not Allowed.

Order Dependence
Within **ASOURCE**: Must precede the **RUN** subcommand.

PCOM	SCOM	input parameters
	FILE	*filename*

filename = (C) Name of file to be written to disk. Typically uses the prefix or suffix REVEXT. No Default.

REF (Reference point) Subcommand
ASOURCE (Acoustic Source) Input Group

Purpose: To specify a reference point which defines the interior and exterior sides of an extrapolation surface.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence
Within **ASOURCE**: Must precede any **POLY**, **CIRC**, **REC** and **RUN** subcommands.

PCOM	SCOM	input parameters
	REF	*xref yref zref refoption*

refoption = (C) Either: IN or OUT. If set to IN, a line drawn from the reference point to the extrapolation surface will touch the inside of the surface. If set to OUT, the line would touch the outside of the surface. The extrapolation process extrapolates the solution outside the surface.

xref, yref, zref = (F) The x- , y- and z-coordinates of the reference point. Default: *xref = 0.0*, *yref = 0.0*, *zref = 0.0*

RESOLUTION Subcommand
ASOURCE (Acoustic Source) Input Group

ASOURCE-RESOLUTION

Purpose: To set the resolution of the grid on the extrapolation surface.

Use Is: Required.

Multiple Use: Allowed. Repeating the subcommand overwrites previously input parameters

Order Dependence
Within **ASOURCE**: Must precede the **RUN** subcommands to which it applies and follow the **FLUID** subcommand.

PCOM	SCOM	input parameters
RESOLUTION		*freqmax nstcyc delx*

freqmax = (F) Maximum frequency of the signal to be extrapolated.
No Default. Must be positive.

nstcyc = (F) Number of time steps per cycle.
Default = 20.

delx = (F) Grid spacing.
Default = *wavespeed* * *dt*.

Notes: 1. Time step is calculated as *dt = 1. / (freqmax * nstcyc)*

POLY Subcommand
ASOURCE (Acoustic Source) Input Group **ASOURCE-POLY**

Purpose: To define a piston with a polygonal cross-section as an acoustic source.

Use Is: Optional. Must define at least one of POLY, CIRC, RECT.

Multiple Use: Allowed. Multiple sources can be defined for each extrapolation file.

Order Dependence
Within **ASOURCE**: Must precede the **RUN** subcommand.

PCOM SCOM input parameters

POLY *axisname drivename scale timeshift x1 y1 x2 y2 x3 y3 ... xN yN*

axisname = (C) Name of previously defined axis system. Default = STND. See Note 1.

drivename = (C) Name of a previously defined driving function. No Default. See Note 2.

scale = (F) Scale factor for driving function. No Default.

timeshift = (F) Time shift on driving function signal. No Default.

x1, y1, ..., xN, yN = (F) Pairs of local coordinate values defining vertices on the polygon cross-section. No Default. See Note 3.

Notes:
1. The axis system defines the orientation of the piston. The local z-axis defines the direction along the length of the piston (the direction of motion) and x-axis and y-axis define the plane of the cross-section.
2. The driving function can be defined using a FUNC or DATA HIST command.
3. Points must be ordered sequentially around the circumference of the polygon. ie. x1, y1 and x2,y2 define the first segment, x2,y2 and x3,y3 define the second segment, etc. A minimum of 3 points must be defined. The maximum number of points allowed is 50.
4. Sources can be superimposed, i.e. multiple sources can occupy the same physical location.

CIRC Subcommand
ASOURCE (Acoustic Source) Input Group **ASOURCE-CIRC**

Purpose: To define a piston with a circular or annular cross-section as an acoustic source.

Use Is: Optional. Must define at least one of POLY, CIRC, REC.

Multiple Use: Allowed. Multiple sources can be defined for each extrapolation file.

Order Dependence
Within **ASOURCE**: Must precede the **RUN** subcommand.

PCOM SCOM input parameters

CIRC *axisname xcent ycent radO radI drivename scale timeshift*

axisname = (C) Name of previously defined axis system. Default = STND. See Note 1.

xcent, ycent = (F) Local coordinates of the circle center. No Default.

radO = (F) Radius of the outer edge of the disk. No Default.

radI = (F) Radius of the inner edge of the annular disk. Set to 0. For a solid circular disk. No Default.

drivename = (C) Name of a previously defined driving function. No Default. See Note 2.

scale = (F) Scale factor for driving function. No Default.

timeshift = (F) Time shift on driving function signal. No Default.

Notes:
1. The axis system defines the orientation of the piston. The local z-axis defines the direction along the length of the piston and x-axis and y-axis define the plane of the cross-section.
2. The driving function can be defined using a FUNC or DATA HIST command.
3. Sources can be superimposed, i.e. multiple sources can occupy the same physical location

RECT Subcommand
ASOURCE (Acoustic Source) Input Group **ASOURCE-RECT**

Purpose: To define a piston with a rectangular cross-section as an acoustic source.

Use Is: Optional. Must define at least one of POLY, CIRC, RECT.

Multiple Use: Allowed. Multiple sources can be defined for each extrapolation file.

Order Dependence
Within **ASOURCE**: Must precede the **RUN** subcommand.

PCOM SCOM input parameters

RECT *axisname xmin xmax ymin ymax drivename scale timeshift*

axisname = (C) Name of previously defined axis system. Default = STND. See Note 1.

xmin, xmax= (F) Local x coordinates of rectangle. No Default.

ymin, ymax = (F) Local y coordinates of rectangle. No Default.

drivename = (C) Name of a previously defined driving function. No Default. See Note 2.

scale = (F) Scale factor for driving function. No Default.

timeshift = (F) Time shift on driving function signal. No Default.

Notes:
1. The axis system defines the orientation of the piston. The local z-axis defines the direction along the length of the piston and x-axis and y-axis define the plane of the cross-section.
2. The driving function can be defined using a FUNC or DATA HIST command.
3. Sources can be superimposed, i.e. multiple sources can occupy the same physical location.

RUN Subcommand
ASOURCE (Acoustic Source) Input Group **ASOURCE-RUN**

Purpose: To run the acoustic source simulation and write the REVEXT data file.

Use Is: Required.

Multiple Use: Not Allowed. Only one RUN subcommand is allowed for each ASOURCE command.

Order Dependence
Within **ASOURCE**: Must follow all other subcommands that specify parameters for this extrapolation file and must be within the same **ASOURCE** command.

PCOM	SCOM	input parameters
	RUN	*pulse_length*

pulse_length = (F) Length of time to generate pressure and gradient data for storage in the REVEXT data file. No default.

AXIS Input Command

AXIS

Purpose: To define local coordinate systems relative to the global Cartesian system. These new local systems can be accessed by other options in order to facilitate transforms between global and local systems. The **AXIS** command group is identical to that described in Section 3. Refer to that section for a description of the options and parameters.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any other command that references the axis definitions.

PCOM	SCOM	input parameters
AXIS		
	DEFN	*axisname axistype x0 y0 z0 x1 y1 z1 x2 y2 z2*
	FORM	*option*
	END	

Note: The subcommands in the **AXIS** command group are identical to the equivalent subcommands of **PZFlex**. See Section 3 for details.

CONVERT Input Command

Purpose: To convert from one file format to another

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM SCOM input parameters

CONVERT

POV *format filename_in filename_out*

END

POV Subcommand
CONVERT Input Group **CONVERT-POV**

Purpose: To convert POV geometry format to an alternate format for reading into an external program.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: None

PCOM	SCOM	input parameters
	POV	*format filename_in filename_out*

format = (C) Set to RAW to convert to Rhino raw triangle format.

filename_in = (C) Name of existing POV file.

filename_out = (C) Name of file to be created in the new format

DATA Input Command

DATA

Purpose: To provide the user with direct IO access to data manager arrays. The **DATA** commands are the same as for PZFlex and are documented in Section 3.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Should be used at the appropriate place in the input stream to read in or write out the data of interest.

PCOM SCOM input parameters

DATA

ABS	*datanameabs dataname1 dataname2 . . . datanamen*
ADD	*datanamefrom scale datanameto ibegin iend jbegin jend kbegin kend*
CDDO	*dataname filename ibegin iend jbegin jend kbegin kend*
CLOS	*dataname*
COPY	*datanamefrom datanameto*
CPYG	*datanamefrom ibgfrom iincfrom jbgfrom jincfrom kbgfrom kincfrom & datanameto ibgto iendto jbgto jendto kbgto kendto*
DIVIDE	*dataname_denom datamane_numer*
EXPONENT	*dataname exponent*
FFT	*realarray imagarray direction inverse*
FFTSHFT	*array1 (array2 array3 ...arrayN) direction*
FILE	*option filename*
HIST	*dataname nvalues filename scaletime scaledata*
IMPORT	THRM *filename matname*
IN	*dataname*
INTERP	*array arraynew*
LOG	*dataname logdataname maxvalue*
MATH	*arraynew* = [algebraic expression involving arrays]

PCOM SCOM input parameters

DATA

MGR	DIM *dataname ndimension ndim1 (ndim2) (ndim3)*
MGR	MAP *dataname itype mapname*
MIRR	*array arraynew axis option ibeg iend jbeg jend kbeg kend*
MULTIPLY	*dataname1 dataname2*
OPEN	*dataname ndimension ndimi ndimj ndimk datatype*
OUT	*dataname*
OUT0	*dataname*
OUT1	*dataname*
PHAD	*ampfrom phasfrom scale ampto phasto ibegin iend jbegin jend kbegin kend*
POLR	*real imag option amp phas ibegin iend jbegin jend kbegin kend*
SCAL	*dataname scalevalue ibegin iend jbegin jend kbegin kend*
END	

Note: The **DATA** subcommands are the same for **Review** as for **PZFlex**. See Section 3 for documentation for the individual subcommands.

DERV (Derivative) Input Command

Purpose: Compute derivative of the specified time histories (i.e., compute acceleration-time histories from velocity-time histories, etc.).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must follow the creation of the data file being differentiated.

PCOM _____ input parameters _____

DERV *filename (label1 label2 . . . labeln)*

filename = (C) Name of file containing time histories whose derivative will be computed.

label1,. . .,labeln = (C) Optional input parameters. Labels of data records to take derivative of (XVEL, SGYY, etc.). If no labels are input, all time histories are differentiated. If labels are input, only data records having the labels specified are differentiated.

Notes: 1. This command replaces the original data contained in the file with its derivative. The original data are no longer available in *filename*.

2. If the user desires to maintain the original data in *filename* for plotting, the **MAKE** option provides an alternative way of creating new time histories that are the derivatives of selected time histories contained in *filename*.

3. Differentiated results can sometimes be very noisy. The **FILT** command may be used to filter out numerical noise.

EXTR (Extrapolate - Kirchhoff) Input Command

Purpose: To extrapolate pressure through a homogeneous acoustic medium using the FLXEXT data file from a **PZFlex** computation. Uses Kirchhoff extrapolation. Produces either time-histories or single frequency results at the requested locations.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM SCOM _____ input parameters _____

EXTR

DATA	*pathname1 pathname2 ....pathnamen*
FILE	*filename (option) fileid*
FREQ	*frequency (attenuation)*
MIRR	*side_normal symm_type* (*side_normal symm_type*) & (*side_normal symm_type*)
SHOW	*display_quantity*
SLVR	*slvrtype*
SURF	*SPHR axisname r begθ endθ nθ begϕ endϕ nϕ*
or	
SURF	*CYLN axisname r zbeg zend nz begθ endθ nθ*
or	
SURF	*QUAD axisname ni nj x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4*
or	
SPOT	*x y z*
TYPE	*option*
VOL	*HEX axisname ni nj x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4…x8 y8 z8*
CALC	*quad_order stor_opt nsection length iprint*
END	

DATA Subcommand
EXTR (extrapolation) Input Group **EXTR-DATA**

Purpose: To specify the file(s) containing data to be extrapolated. This is typically a file with the FLXEXT identifier.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence within **EXTR** Group: Must precede any **CALC** subcommands to which it applies.

PCOM SCOM input parameters

DATA *pathname1 pathname2 ....pathnamen*

pathname = (C) Pathname(s) of file(s) containing extrapolation data. If restarts were used in generating the extrapolation data, the file names should be entered in sequential order. Otherwise a single pathname should be entered.
No Default.

FILE Subcommand
EXTR (extrapolation) Input Group **EXTR-FILE**

Purpose: To define a new directory name that will contain extrapolation results.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence within EXTR Group: Must precede any **CALC** subcommands to which it applies.

PCOM	SCOM	input parameters
	FILE	*filename (option) fileid*

filename= (C) Name of new local file that will contain time histories created during time domain extrapolation or the directory name for frequency domain results. See Note 1 below. For frequency domain results, the default directory is extr.

option= (C) Optional input. Set to CLER to clear all records previously stored in local file *filename*.

fileid= (C) Used only for TIME domain extrapolation.. The identifier (up to 8 characters) that will be associated with file *filename*. This identifier affects curve labels when curves with different identifiers are cross plotted. The identifier is associated with the file if it is written to a disk file with the **WRIT** command.

Notes: 1. The time histories are stored in standard time history form with a unique time record corresponding to each data record. To cross plot the first 2 records on an extrapolated data file f1, use the commands:

```
GRPH
    PLOT f1 1 2
```

2. For frequency domain, the syntax is:

```
GRPH
    PLOT extr/pf:m1
```

FREQ (frequency) Subcommand
EXTR (extrapolation) Input Group **EXTR-SLVR**

Purpose: To specify a frequency for frequency domain extrapolation.

Use Is: Required for frequency domain solver.

Multiple Use: Allowed.

Order Dependence within **EXTR** Group: Must precede any **CALC** subcommands to which it applies.

PCOM SCOM input parameters

FREQ *frequency (attenuation)*

frequency = (F) Extrapolation frequency [cycles / unit time].
Default = 0.

attenuation = (F) Amplitude attenuation [dB / unit distance].
Default = 0.

MIRR (Mirror) Subcommand
EXTR (extrapolation) Input Group **EXTR-MIRR**

Purpose: To mirror data prior to processing. Typically used with data where symmetry conditions were used in the simulation to calculate a half, quarter or eighth model.

Use Is: Optional. If not input, no symmetry conditions are assumed.

Multiple Use: Allowed.

Order Dependence within **EXTR** Group: Must precede any **CALC** subcommands to which it applies.

PCOM SCOM input parameters

MIRR *side_normal symm_type* (*side_normal symm_type*) (*side_normal symm_type*)

side_normal = (C) Outward normal direction for a side that is a symmetry plane
Any of: X-, X+, Y-, Y+, Z-, Z+, or OFF
If set to OFF, all mirroring is turned off.
No default.

symm_type= (C) Symmetry type.
Any of: SYMM = symmetric response
Default = SYMM.

Notes:
1. From one to three symmetry planes may be defined using the **MIRR** subcommand.
2. Only one symmetry plane may be defined for each axis.
3. Only model symmetry planes that intersect data planes defined by the **EXTR NODE** command in the **PZFlex** analysis must be specified.

SHOW Subcommand
EXTR (extrapolation) Input Group **EXTR-SHOW**

Purpose: To save information about the extrapolated pressure-time histories at each extrapolated spatial location for future display. Applies to time domain extrapolation only.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **EXTR** Group: Must precede any **CALC** subcommand to which it applies.

PCOM SCOM input parameters

SHOW *display_quantity*

display_quantity= (C) Display quantity to extract at each observation point.
Any of: PEAK = Peak amplitude (absolute value)
PTOP = Peak-to-peak amplitude
NONE = No display quantities
at each observation point. Default = PEAK

Note: These quantities are stored in the EXTR/PEAK or EXTR/PTOP arrays respectively. They may be plotted using standard GRPH commands.

SLVR (Solver) Subcommand
EXTR (extrapolation) Input Group **EXTR-SLVR**

Purpose: To request TIME domain or FREQUENCY domain extrapolation.

Use Is: Optional, TIME assumed if not input.

Multiple Use: Not allowed.

Order Dependence within **EXTR** Group: Must precede any **CALC** subcommands to which it applies.

PCOM	SCOM	input parameters
	SLVR	*slvrtype*

slvrtype = (C) Extrapolation type. Either TIME for time domain or FREQ for frequency domain.
Default = TIME.

SURF (Surface) Subcommand
EXTR (Extrapolate) Input Group **EXTR-SURF**

Purpose: To specify 1D curves or 2D surfaces of observation points for which the pressure field will be extrapolated. Curves are specified as degenerate surfaces, e.g., a cylinder with nz=1, or a quad with nj=1.

Use Is: Either SURF or SPOT is required.

Multiple Use: Allowed.

Order Dependence
within **EXTR** Group: Must preceed related CALC subcommand.

PCOM SCOM input parameters

SURF *SPHR axisname r begθ endθ nθ begϕ endϕ nϕ*
or
SURF *CYLN axisname r zbeg zend nz begθ endθ nθ*
or
SURF *QUAD axisname ni nj x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4*

axisname = (C) The name of a local coordinate system defined with the **AXIS** primary command. The local system should be cylindrical for a CYLN surface, spherical for a SPHR surface or Cartesian for a QUAD surface. Default = global system.

r = (C) Radius Default: None.

begθ endθ = (F) Beginning and ending angles (degrees) in the specified local coordinate system. No default.

nθ = (I) Number of points in the θ direction. No default.

begϕ endϕ = (F) Beginning and ending angles (degrees) in the specified local coordinate system. No default.

nϕ = (I) Number of points in the ϕ direction. No default.

zbeg, zend= (F) Beginning and ending coordinates of cylinder axis in the specified local coordinate system. No default.

nz = (I) Number of points in the z direction. No default.

ni, nj = (I) Number of points in the i and j local directions respectively.

(x1,y1,z1) ...(x4,y4,z4)= (F) Corner coordinates of a general quadrialteral in the local system. Points 1 and 3 are located at minimum i, points 1 and 2 are located at minimum j.

SPOT Subcommand
EXTR (Extrapolate) Input Group **EXTR-SPOT**

Purpose: To specify a single observation point to which the pressure field will be extrapolated.

Use Is: Either SURF or SPOT is required.

Multiple Use: Allowed.

Order Dependence within **EXTR** Group: Must preceed related CALC subcommand.

PCOM	SCOM	input parameters
	SPOT	*x y z*

x, y , z= (F) Coordinates in global directions of model.
Default = (0.,0.,0.)

VOL (Volume) Subcommand
EXTR (Extrapolate) Input Group **EXTR-VOL**

Purpose: To specify a 3D volume of observation points for which the pressure field will be extrapolated.

Use Is: Either VOL, SURF or SPOT is required.

Multiple Use: Allowed.

Order Dependence
Within **EXTR** Group: Must preceed related CALC subcommand.

PCOM SCOM input parameters

VOL *HEX axisname ni nj x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4…x8 y8 z8*

axisname = (C) The name of a local coordinate system defined with the **AXIS** primary command. The local system should be Cartesian for a HEX region. Default = global system.

ni, nj,nk = (I) Number of points in the i, j and k local directions respectively.

(x1,y1,z1) ...(x8,y8,z8)= (F) Corner coordinates of a general hexahedron in the local system. Points 1, 3, 5, 7 are located at minimum i, points 1, 2, 5,6 are located at minimum j; points 1,2,3,4 are at minimum k.

TYPE Subcommand
EXTR (extrapolation) Input Group **EXTR-SLVR**

Purpose: To choose REAL/IMAG or AMP/PHASE results from frequency domain extrapolation.

Use Is: Optional, REAL assumed if not input.

Multiple Use: Not allowed.

Order Dependence within **EXTR** Group: Must precede any **CALC** subcommands to which it applies.

PCOM SCOM input parameters

TYPE *option*

option = (C) Result type. Either REAL or AMP
Default = AMP.

Notes:
1. For option = REAL, the real and imaginary parts of the extrapolated pressure are stored in arrays pf:r*n* and pf:i*n*, respectively, where *n* is the frequency number. For type = AMP, the amplitude and phase of the extrapolated pressure are stored in pf:m*n* and pf:p*n* where *n* is the frequency number. Each of these arrays is dimensioned (n1,n2,ns) where n1 is the the number of points in the first direction of the output surface, n2 is the number of points in second direction of the output surface, and ns is the number of output surfaces (see SURF subcommand).
2. Only one symmetry plane may be defined for each axis.
3. Only model symmetry planes that intersect data planes defined by the **EXTR NODE** command in the **PZFlex** analysis must be specified.

CALC (calculate) Subcommand
EXTR (extrapolation) Input Group **EXTR-CALC**

Purpose: To perform the extrapolation.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **EXTR** Group: Must follow the **FILE, DATA, SURF,** and any **MIRR, TIME** or **SHOW** subcommands wthat apply to this extrapolation computation.

PCOM SCOM input parameters

CALC *quad_order stor_opt nsection length iprint*

or

quad_order ARRAY tablename

quad_order= (I) Gaussian quadrature order. Number of integration points in each direction per element. Default = 1.

stor_opt= (C) Option to store or calculate Green's functions for 2D extrapolation of plane strain models. Either STOR or CALC. CPU time can be dramatically reduced by storing them, but the array can become quite large. The size of the array required for the STOR option is equal to (2 * *nsteps* * *nelem*) words where *nsteps* is the number of timesteps on the data file, and *nelem* is the number of elements in the data surface. Default = STOR.

nsection (I) Used only for 2.5D (see notes 1 and 2 below) and axisymmetric models. The integration in z or theta is performed using *nsection* discrete points. Default = 36 for axisymmetric, *length / (c * timestep)* for 2.5D.

length (F) Used only for 2.5D models. Length in the z-direction. Default for time domain = infinite length. Length must be input for frequency domain extrapolation from 2D plane strain models.

iprint (I) Enter OFF to suppress echo printing of extrapolation points to the flxprt file. Default = ON.

tablename (C) Name of a previously defined data array containing space and time shifts. See note 3.

(continued next page)

Notes:
1. By 2.5D we mean 3D extrapolation from a 2D finite element model. This provides an approximate method to account for the finite length in the z-direction.
2. The 2.5D and axisymmetric extrapolation options generate 3D data sets by replicating the 2D calculated results. For near field extrapolation, the discretization in the z or theta directions should be about the same size as a finite element in the model. In the far field, this can be coarsened by a large factor. A rule of thumb is that the difference in travel times from adjacent segments to the observation point should be about (*c*timestep*). A simple trial and error method is to pick a few bounding points on the extrapolation surface and decrease *nsection* as much as possible without changing the solution.
3. The ARRAY option (3D models only) creates duplicate copies of the extrapolation file data shifted in space and time. This provides a means to approximate an entire device based on a simulation of a smaller portion of the device. DATA TABL or DATA INTEXT provide convenient means to create the *tablename* array. For each copy of the data, tablename(i,1) is the x −shift, tablename(i,2) is y-shift, tablename(i-3) is z-shift and tablename(i,4) is time shift. Include (0,0,0,0) for the baseline copy.

FILT (Filter) Input Command

Purpose: To filter time histories. Three different filters are available.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the commands that define the time history records for a file.

PCOM _____ input parameters _____

FILT *filename lowfilter highfilter*

or

filename DIG *fcut fband*

or

filename FFT *freqwndo timewndo*

filename = (C) Name of file to be filtered. All records on the file are filtered. Note: options on the READ command permit a subset of a disk file to be read onto *filename*.

lowfilter = (F) Low-frequency limit (cycles/unit time) of the bandpass window. At the frequency *lowfilter,* the amplitude is reduced by 3 dB (a factor of 2)

highfilter = (F) High-frequency limit (cycles/unit time) of the filter window. At the frequency *highfilter,* the amplitude is reduced by 3 dB (a factor of 2)

fcut = (F) Cutoff frequency (cycles/unit time) in problem units. No default. See note 3 below.

fband = (F) Frequency band in problem units. No default.

freqwndo = (C) Name of a frequency domain window function previously defined via a DATA HIST command. Default = none. See note 4 below.

timewndo = (C) Name of a time-domain window function previously defined via a DATA HIST sommand. Default = none.

FILT

Notes: 1. The default filter is a 3rd order-recursive Butterworth filter. The method impelments a two pass, phaseless filter that is a computationally efficient approximation to an ideal bandpass filter. Each time history is corrected to have a zero mean before filtering.
The filter procedure writes three numbers to the current output. The first two values, TEST Z1 ROOT and TEST Z2 ROOT, should be close to zero. The third value,
TEST GAIN ESTIMATE, should be close to 1. If not, the filtered results are not valid.

2. The original version of this filter procedure was written by J. Bernard Minster at Cal Tech in 1979. It is based on a procedure outlined by C. M. Rader and B. Gold in "Digital Filter Design Techniques in the Frequency Domain," *Proc. IEEE*, 55,
pp. 149-171, 1967.
3. The DIG option invokes a digital lowpass filter. 95 percent is passed at *fcut-fband/2* and 5% is passed at *fcut+fband/2*. This filter does not require zero mean or equal end values, but the starting value must be zero. It cannot process the end of the record, which it sets to zero
4. The FFT option implements filtering via FFTs. Each record is scaled by the *timewndo* function before processing. Because FFTs imply periodicity, the time window should be applied to bring both ends to zero (or at least the same constant value). The frequency window provides a general method for filtering.

FREQ (frequency) Input Command

FREQ

Purpose: To transform time domain data into frequency domain data by performing an FFT on specified time histories. The amplitude and phase data are accumulated as curves on a local file and may be cross-plotted with other data.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Should follow the commands that read in the base time histories used to compute the FFTs.

PCOM SCOM — input parameters

FREQ

FILE	*fftfilename fileid*
TTL	
<----------60 character title for *fftfilename* ---------->	
TIME	FROM *filename*
or	
TIME	*ntime timebegin timeinc*
FFT	*filename icurve*
NFFT	[*record1*] [*record2*]
SFRQ	*freqbegin freqend nfreq*
SPEC	*filename record dampratio*
TYPE	*option*
IMPD	[*curvevolt*] [*curvecharge*] *scalevolt scalecharge*
ADMT	[*curvevolt*] [*curvecharge*] *scalevolt scalecharge*
WNDO	*type option fraction*
ZIMP	[*curvepres*] [*curvevel*] *density wavespeed*
END	

Example 1: An example of a complete **Review** input file using the **FREQ** input command group to calculate the FFT of a time history and plot the fourier amplitude vs. frequency curve is as follows.

```
READ      F1   FLXHST.RESULTS     /* read in base time histories
FREQ
  FILE    F2        /* define a new local file name for new records
  FFT     F1  12    /* compute FFT of record 12 on file F1
  FFT     F1  21    /* compute FFT of record 21 on file F1
  C       each FFT command adds two records to file F2
  C       the first contains fourier amplitude data and the second contains phase data
  END
GRPH
C cross-plot amplitude vs. frequency data from the two FFTs computed
  PLOT    F2  1  F2  3
  PS                /* create PostScript image
  END
```

FILE Subcommand
FREQ (frequency) Input Group **FREQ-FILE**

Purpose: To define a new local file name which will contain the frequency domain data computed with **FFT** subcommands.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must precede any **TIME** and **FFT** subcommands used to compute the frequency domain data to be contained in *fftfilename*.

PCOM	SCOM	input parameters
	FILE	*fftfilename (option) fileid*

fftfilename = (C) Name of new local file that will contain time histories created by **FFT** subcommands.

option = (C) Optional input parameter. If set to CLER, all previous curves stored in *fftfilename* will be deleted.

fileid = (C) Identifier (up to 8 characters) that is associated with the file *fftfilename*. This can also be set or modified with the **ID** command. This identifier affects curve labels when curves with different identifiers are cross-plotted. The identifier is associated with the file if it is written to a disk file with the **WRIT** command. If not input, *fileid* defaults to that of the file providing the time history used by the first **FFT** subcommand encountered.

Note: Each **FFT** subcommand that follows a **FILE** subcommand adds two new data curves to *fftfilename*. If a new **FILE** subcommand is encountered, all subsequent **FFT** subcommands add two curves to the new *fftfilename*.

TTL (title) Subcommand
FREQ Input Group **FREQ-TTL**

Purpose: To provide a title for the current *fftfilename*. This title will be used only if *fftfilename* is written to an external disk file with the **WRIT** command. Otherwise, this command can be ignored.

Use Is: Optional. If not input, the file title will default to that of the file providing the first time history used by the first **CURV** subcommand encountered.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	TTL	<----------60 character title for *fftfilename* ---------->

Note: The title line must be input immediately after the **TTL** subcommand line.

TIME Subcommand
FREQ (frequency) Input Group **FREQ-TIME**

Purpose: To define the time sampling for the file *fftfilename* created with the last **FILE** subcommand.

Use Is: Optional. If not input, the time information for *fftfilename* defaults to that of the time history used by the first subcommand encountered that computes an FFT.

Multiple Use: Allowed. One use for each **FILE** subcommand.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand to which it applies and precede all **FFT** subcommands that add amplitude and phase data to *fftfilename*.

PCOM SCOM input parameters

<< use time information from another existing local time history file>>
TIME FROM *filename*

filename = (C) If the first input parameter is FROM, *filename* defines another local file from which time sampling information is taken.

or

<< use time information from first time record used to compute FFT - allow padding>>
TIME PAD *npad time_phase0 inc_scale*

npad = (I) Padding factor for the baseline time record. *npad*=2 indicates that the number of time points is doubled, the time increment remains the same, and the overall time record length is doubled. Data values outside of the original records time range are set to zero.
Default = 1

time_phase0 = (F) Option to establish the reference time representing zero phase.
Default = 0.0

inc_scale = (I) Integer scale factor on the time increment. *inc_scale*= 2 indicates that every second data point in an equal-spaced time spaced record is removed before the FFT is computed.
Default = 1

or

<< user-specified time record to use for computing FFTs for *fftfilename*>>
TIME *ntime timebegin timeinc*

ntime = (I) Number of data-time pairs to use to compute FFT data to be added to *fftfilename*.

timebegin = (F) Beginning time of the time histories.

timeinc = (F) Time increment between time sampling points. The ending time , *timeend*, of the time histories is:
timeend = *timebegin* + *timeinc* x (*ntime* - 1)

Note: It is often possible to default the time information for a new time history file and skip the use of the **TIME** subcommand to explicitly set the time information.

FFT (fast Fourier transform) Subcommand
FREQ (frequency) Input Group **FREQ-FFT**

Purpose: To compute the FFT of a time history and accumulate the amplitude and phase information to *fftfilename*.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	FFT	*filename record*

filename = (C) Local file name containing the record to use. No default.

record = (I) Record number on the file. No default.

Notes:
1. Each **FFT** subcommand adds two additional data records to *fftfilename*. The first contains the computed fourier amplitudes. The second contains phase information.
2. New data curves created using the **FFT** subcommand have default curve labels. When plotting these curves, the user can specify desired curve labels using the **GRPH SET** subcommand.
3. The **FFT** command computes the FFT for equal-spaced time data that are zero padded out to the nearest power of 2. If the time history whose FFT is being computed does not begin and end at 0.0, the padding causes a drop from its last value to zero over one time step. The resulting FFT reflects this artificial portion of the time history record. The **WNDO** subcommand can be used to ensure that the record smoothly transitions to zero over its time range.
4. The spacing between frequency values that are evaluated using the **FFT** command is the inverse of the total length of the time record being evaluated. To refine the frequency sampling, increase the length of the time record by using the **TIME** subcommand to define a longer time record.
5. FFT computes the Fourier Series Coefficients G^k such that the time record is synthesized from the positive frequencies as: $g(kT) = \mathrm{Re}\left[\sum_{n=0}^{N/2-1} G^k e^{\frac{i2\pi nk}{N}}\right]$.

   The Fourier Transform coefficients are obtained as $G^k/\Delta f$ where Δf is the frequency spacing (the frequencies are in the array *fftfilename*/TIME).
6. The command **SYMB #** GET {DELF} CURVDELX *fftfilename* assigs the frequency spacing to variable DELF.

NFFT (Normalized FFT) Subcommand
FREQ (frequency) Input Group **FREQ-NFFT**

Purpose: To compute the FFT of a time history record normalized by the FFT of a second time history record. This command replaces having to use multiple **FREQ** and **MAKE** commands to normalize plots, e.g., for determining displacement per volt.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand that sets up the local file to contain the time history curves.

PCOM SCOM input parameters

NFFT [*record1*] [*record2*]

record1 = (X) Time history record specification to be normalized. Composed of a file name and record number (i.e., F1 3) or just a record number (i.e., 6). If no file name is provided, the first file read into Review is assumed. No default.

record2 = (X) Time history record specification of the record used for normalizing *record1*. Composed of a file name and record number (i.e., F1 3) or just a record number (i.e., 6). No default.

Note: Each **NFFT** command adds two frequency records to the current frequency file. The records are either amplitude and phase (in degrees) or real and imaginary components, depending on the form of the last **TYPE** subcommand input. If no **TYPE** subcommand was input, the default is amplitude and phase.

SFRQ (spectra frequencies) Subcommand
FREQ (frequency) Input Group **FREQ-SFRQ**

Purpose: To generate a special frequency table to be used for response spectra calculations.

Use Is: Optional.

Multiple Use: Allowed only one for each **FILE** subcommand.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand and precede any **SPEC** or **TIME** subcommands.

PCOM	SCOM	input parameters
	SFRQ	*freqbegin freqend nfreq*

freqbegin = (F) Beginning frequency to be calculated.
Must be greater than 0.0.
No default.

freqend = (F) Ending frequency to be calculated.
Must be greater than 0.0.
No default.

nfreq = (I) Number of frequencies to evaluate.
Default = 100.

Notes: 1. If this option is used to create the frequency vector to be evaluated for local file *freqfilename*, the FFT subcommand may not be used to add records to *freqfilename*.

2. This option generates a frequency list spanning from *freqbegin* to *freqend* in (*nfreq*-1) steps, where:

f(1)=*freqbegin*
f(2)=f(1)**freqratio*
f(3)=f(2)**freqratio*
.
.
.
f(*nfreq*)=f(*nfreq*-1)**freqratio*=*freqend*

and

$$freqratio = \left(\frac{freqend}{freqbegin}\right)^{\left(\frac{1}{nfreq-1}\right)}$$

SPEC (response spectra) Subcommand
FREQ (frequency) Input Group **FREQ-SPEC**

Purpose: To compute the response spectra of a velocity-time history and accumulate the pseudo displacement, velocity and acceleration response records to the file *freqfilename*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	SPEC	*filename record dampratio*

filename = (C) Local file name containing the record to use.
No default.

record = (I) Record number on the file.
No default.

dampratio = (F) Damping ratio to use in computing the response spectra.
Default=.02 (2% damping)

Notes: 1. Each **SPEC** subcommand adds three additional data records to *freqfilename*. The first contains the computed pseudo-displacement data, the second contains psedo-velocity, and the third is pseudo-acceleration.

2 New data curves created using the **SPEC** subcommand have default curve labels. When plotting these curves, the user can specify any desired curve labels using the **GRPH SET** subcommand.

TYPE Subcommand
FREQ (frequency) Input Group **FREQ-TYPE**

Purpose: To select the type of impedance or FFT curves to compute when using the **IMPD**, **ADMT,** or **FFT** subcommands; either real and imaginary or amplitude and phase options may be chosen.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Affects the action for **IMPD**, **ADMT,** and **FFT** subcommands that follow.

PCOM SCOM input parameters

TYPE *option*

option = (C) Either: REAL = compute real and imaginary curves
AMP = compute amplitude and phase curves

IMPD (Impedance) Subcommand
FREQ (frequency) Input Group **FREQ-IMPD**

Purpose: To compute the complex impedance using time histories of voltage and charge. The charge time history is automatically differentiated to produce the current time history required for computing impedance.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand that sets up the local file to contain the impedance curves.

PCOM SCOM input parameters

IMPD [*curvevolt*] [*curvecharge*] *scalevolt scalecharge impopt*

curvevolt = (X) Time history record specification of the voltage on the electrode of interest. Composed of a file name and record number (i.e., F1 3) or just a record number (i.e., 6). If no file name is provided, the first file read into Review is assumed. No default.

curvecharge = (X) Time history record specification of the charge on the electrode of interest. Composed of a file name and record number (i.e., F1 3) or just a record number (i.e., 6). If no file name is provided, the filename for *curvevolt* is assumed. No default.

scalevolt = (F) Factor to scale the voltage before computing the impedance. Default = 1.0

scalecharge = (F) Factor to scale the charge before computing the impedance. Default = 1.0

impopt= (C) Impedance calculation option. Any of: NUMR or ANAL.
NUMR = numerical differentiation of charge
ANAL = analytic differentiation of charge
Default = NUMR. See note 2.

Notes:
1. Each **IMPD** command adds two frequency records to the current frequency file. The records are either amplitude and phase (in degrees) or real and imaginary components, depending on the form of the last **TYPE** subcommand input. If no **TYPE** subcommand was input, the default is amplitude and phase.
2. The **IMPD** command uses an FFT approach to computing admittance. The FFT is computed for equal-spaced time data that are zero padded out to the nearest power of 2. If the time histories whose FFT is being computed does not begin and end at 0.0, this padding causes a drop from its last value to zero over one time step. The resulting impedance reflects this artificial portion of the time history record. The **WNDO** subcommand can be used to ensure that the records smoothly transition to zero over their time range.
3. The spacing between frequency values that are evaluated using the **ADMT** command is the inverse of the total length of the time record being evaluated. To refine the frequency sampling, increase the length of the time record by using the **TIME** subcommand to define a longer time record.
4. For *impopt* = NUMR, the charge is differentiated numerically to calculate current then FFT'd. This is the preferred method, but can be inacurrate if the **POUT RATE** is too coarse. For *impopt* = ANAL, the charge is FFT'd, and the current is obtained analytically. For this method, the charge should be equal at the beginning and end of the calculation.

Example:

```
READ F1 FLXHST. RUN1 /* read in time history file
FREQ
        FILE F2     /* assigns a local filename to the file that contains impedance curves
        IMPD F1 5 10   /* computes impedance using record 5 for voltage and 10 for charge
GRPH
        NVEW 2          /* selects two views for plotting
        PLOT F2 1       /* plots amplitude vs. frequency
        PLOT F2 2       /* plots phase vs. frequency
C       SET WNDO command may be used to window the frequency range of interest
```

ADMT (admittance) Subcommand
FREQ (frequency) Input Group **FREQ-ADMT**

Purpose: To compute the complex admittance using time histories of voltage and charge. The charge time history is automatically differentiated to produce the current time history required for computing admittance.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must follow the **FILE** subcommand that sets up the local file to contain the admittance curves.

PCOM SCOM input parameters

ADMT [*curvevolt*] [*curvecharge*] *scalevolt scalecharge impopt*

curvevolt = (X) Time history record specification of the voltage on the electrode of interest. Composed of a file name and record number (i.e., F1 3) or just a record number (i.e., 6). If no file name is provided, the first file read into Review is assumed. No default.

curvecharge = (X) Time history record specification of the charge on the electrode of interest. Composed of a file name and record number (i.e., F1 3) or just a record number (i.e., 6). If no file name is provided, the filename for *curvevolt* is assumed. No default.

scalevolt = (F) Factor to scale the voltage before computing the impedance. Default = 1.0

scalecharge = (F) Factor to scale the charge before computing the impedance. Default = 1.0

impopt= (C) Impedance calculation option. Any of: NUMR or ANAL.
NUMR = numerical differentiation of charge
ANAL = analytic differentiation of charge
Default = NUMR. See note 2.

Notes:
1. Each **ADMT** command adds two frequency records to the current frequency file. The records are either amplitude and phase (in degrees) or real and imaginary components, depending on the form of the last **TYPE** subcommand input. If no **TYPE** subcommand was input, the default is amplitude and phase.
2. The **ADMT** command uses an FFT approach to computing admittance. The FFT is computed for equal-spaced time data that zero padded out to the nearest power of 2. If the time histories whose FFT is being computed does not begin and end at 0.0, this padding causes a drop from its last value to zero over one time step. The resulting admittance reflects this artificial portion of the time history record. The **WNDO** subcommand can be used to ensure that the records smoothly transition to zero over their time range.
3. The spacing between frequency values that are evaluated using the **ADMT** command is the inverse of the total length of the time record being evaluated. To refine the frequency sampling, increase the length of the time record by using the **TIME** subcommand to define a longer time record.
4. For *impopt* = NUMR, the charge is differentiated numerically to calculate current then FFT'd. This is the preferred method, but can be inacurrate if the **POUT RATE** is too coarse. For *impopt* = ANAL, the charge is FFT'd, and the current is obtained analytically. For this method, the charge should be equal at the beginning and end of the calculation.

Example:

```
READ F1 FLXHST. RUN1 /* reads in time history file
FREQ
        FILE F2     /* assigns a local filename to the file that contains impedance curves
        ADMT F1 5 10   /* computes admittance using record 5 for voltage and 10 for charge
GRPH
        NVEW 2          /* selects two views for plotting
        PLOT F2 1       /* plots amplitude vs. frequency
        PLOT F2 2       /* plots phase vs. frequency
C       SET WNDO command may be used to window the frequency range of interest
```

WNDO (window) Subcommand
FREQ (frequency) Input Group **FREQ-WNDO**

Purpose: To define a windowing function to scale a time history before computing FFTs for the **FFT**, **IMPD,** or **ADMT** subcommands. This option allows computing of useful FFT results for data that are non zero at either end of the time history. The **WNDO** subcommand applies to all following **FFT**, **IMPD,** and **ADMT** subcommands until it is deactivated.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must precede any **FFT**, **IMPD,** or **ADMT** subcommand for which the windowing function is desired.

PCOM SCOM input parameters

WNDO *type option fraction windobegin windoend*

type = (C) Type of window function to use. See illustration.
Any of: BART = Bartlett window
WELC= Welch window
HANN = Hann window
OFF = Deactivates windowing function
Default = BART

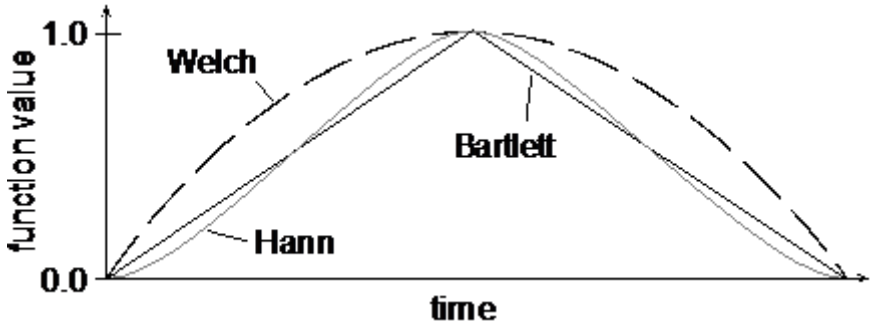
option = (C) Window option. See illustration.
Any of: FULL = Full window
LEFT= Left side of window
RIGH = Right side of window

Default = FULL

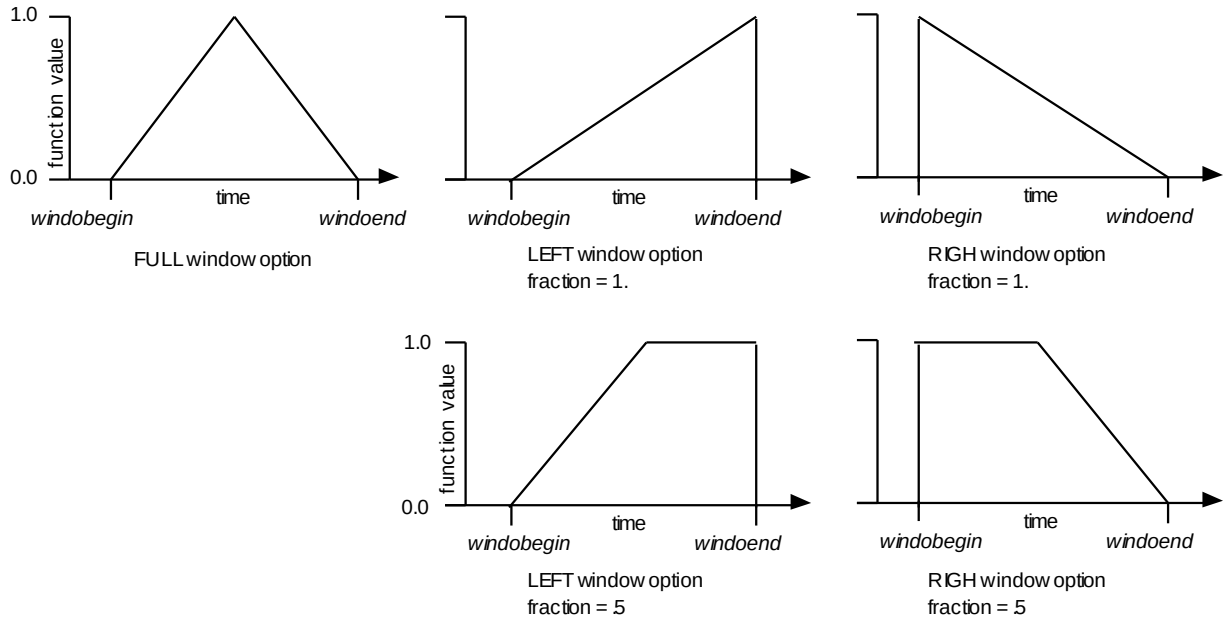
fraction = (F) Fraction of the data range defined by *windobegin* and *windoend* to apply the LEFT or RIGH window option to. *fraction* is a value between 0. and 1. See illustration.
Default = 1.0 means the scaling function varies over the entire time range specified.

windobegin= (F) Value of the window scaling function ranges from 0.0 to 1.0 in the data range *windobegin* to *windoend.* The value of the scaling function outside this range is 0.0. *windobegin* defines the begining time value to which a nonzero scaling value can be applied.
Default = The starting time of the record.

windoend= (F) *windoend* defines the ending time to which a nonzero scaling value may be applied.
Default = The ending time of the record.



Various windowing function types for the FULL window option



Various options for the Bartlett window function

Notes: 1. The function of this command is to scale the data values of a time history by the values of the window function at each corresponding time.

2. These window functions are discussed in the text *Numerical Recipes in FORTRAN, the Art of Scientific Computing - Second Edition* by W. H. Press, S. A. Teukolsky W. T. Vettering and B. P Flannery, Cambridge University Press, 1986.

ZIMP (Acoustic Impedance) Subcommand
FREQ (Frequency) Input Group **FREQ-ZIMP**

Purpose: To compute the complex impedance using time histories of average pressure and velocity.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **FREQ** Group: Must follow the **FILE** subcommand which sets up the local file to contain the impedance curves.

PCOM SCOM input parameters

ZIMP [*curvepres*] [*curvevel*] *density wavespeed*

curvepres = (X) The time history record specification of the pressure on the surface of interest. Composed of a file name and record number (i.e. F1 3), or just a record number (i.e. 6). If no file name is provided, the first file read into REVIEW is assumed. No default.

curvevel = (X) The time history record specification of the velocity on the surface of interest. Composed of a file name and record number (i.e. F1 3), or just a record number (i.e. 6). If no file name is provided, the filename for *curvepres* is assumed. No default.

density= (F) Enter material density to normalize by characteristic impecance. Default = 1.0

wavespeed = (F) Enter material wavespeed to normalize by characteristic impedance. Default = 1.0

(description of **ZIMP** subcommand continued on next page)

ZIMP (Acoustic Impedance) Subcommand
FREQ (Frequency) Input Group

FREQ-ZIMP
(continued)

1. Each **ZIMP** command adds two frequency records to the current frequency file. The records are either amplitude and phase (in degrees), or real and imaginary components depending on the form of the last **TYPE** subcommand input. If no **TYPE** subcommand was input, the default is amplitude and phase.
2. The **ZIMP** command uses an FFT approach to computing admittance. The FFT is computed for equal spaced time data which is zero padded out to the nearest power of 2. If the time histories whose FFT is being computed does not begin and end at 0.0, this padding will cause a drop from its last value to zero over one time step. The resulting impedance will reflect this artificial portion of the time history record. The **WNDO** subcommand can be used to enforce that the records smoothly transition to zero over their time range.
3. The CALC BWORK subcommand in Flex computes the integrals of pressure and velocity over user specified surfaces. These are stored in ZIMP(2,i) and ZIMP(1,i) respectively for surface i.

Example:

```
READ F1 FLXHST. RUN1 /* read in time history file
FREQ
        FILE F2     /* assign a local filename to the file which will contain impedance curves
        ZIMP F1 5 10   /* compute impedance using record 5 for voltage and 10 for charge
GRPH
        NVEW 2          /* select two views for plotting
        PLOT F2 1       /* plots amplitude vs frequency
        PLOT F2 2       /* plots phase vs frequency
C       SET WNDO command may be used to window the frequency range of interest
```

GRPH (graphics) Input Command

GRPH

Purpose: To provide interactive graphics display of time histories on systems supporting the X11 graphics protocol. Also, provides for interactive or batch-mode generation of PostScript image files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the commands that import the time histories to be displayed.

PCOM SCOM input parameters

GRPH

ARAT	*aspectratio*
CLOS	
LIST	*filename (label) (indxlabel index) (nbegin nend ninc)*
NVEW	*nview*
PLOT	*[curve 1] [curve 2] . . . [curve 6]*
PS	
SET/PSET	*option parameters*
SIZE	*xsize ysize xbegin xend ybegin yend nxdiv nydiv*
TTL	*ititle* (FROM *localfile*)

----------200 character title information for line = *ititle*---------->

VIEW	*iview*
WRIT	(*options*) *filename*
END	

ARAT (aspect ratio) Subcommand
GRPH (graphics) Input Group **GRPH-ARAT**

Purpose: To override the default aspect ratio (horizontal/vertical) of 2D or 3D field variable or geometry plots.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Must precede any **PLOT** commands for which the user wishes to control the aspect ratio of the plotted image.

PCOM	SCOM	input parameters
	ARAT	*aspectratio*

aspectratio = (X) Aspect ratio (horizontal/vertical) of all following 2D or 3D plots of field variables or model geometry. This option overrides the internal defaults of the code that are based on actual model geometry or array dimensions. If *aspectratio* = 1.0, the plotted image is a square while if it is set to 2.0, the image is twice as wide as it is tall. If set to 0.0 or OFF, then the internal defaults will be used to display the image.

CLOS (close) Subcommand
GRPH (graphics) Input Group **GRPH-CLOS**

Purpose: To close the graphics window.

Use Is: Optional. If not input, the graphics window is closed at the end of the job.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	CLOS	-none-

LIST Subcommand
GRPH (graphics) Input Group **GRPH-LIST**

Purpose: To provide the functionality of the **LIST** primary command within the **GRPH** command group.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	LIST	*filename (label) (indxlabel index) (nbegin nend ninc)*

See the **LIST** primary command for a description of the input parameters.

NVEW (number of views) Subcommand
GRPH (graphics) Input Group **GRPH-NVEW**

Purpose: To set the number of viewing windows to present on the screen at one time.

Use Is: Optional.

Multiple Use: Allowed.

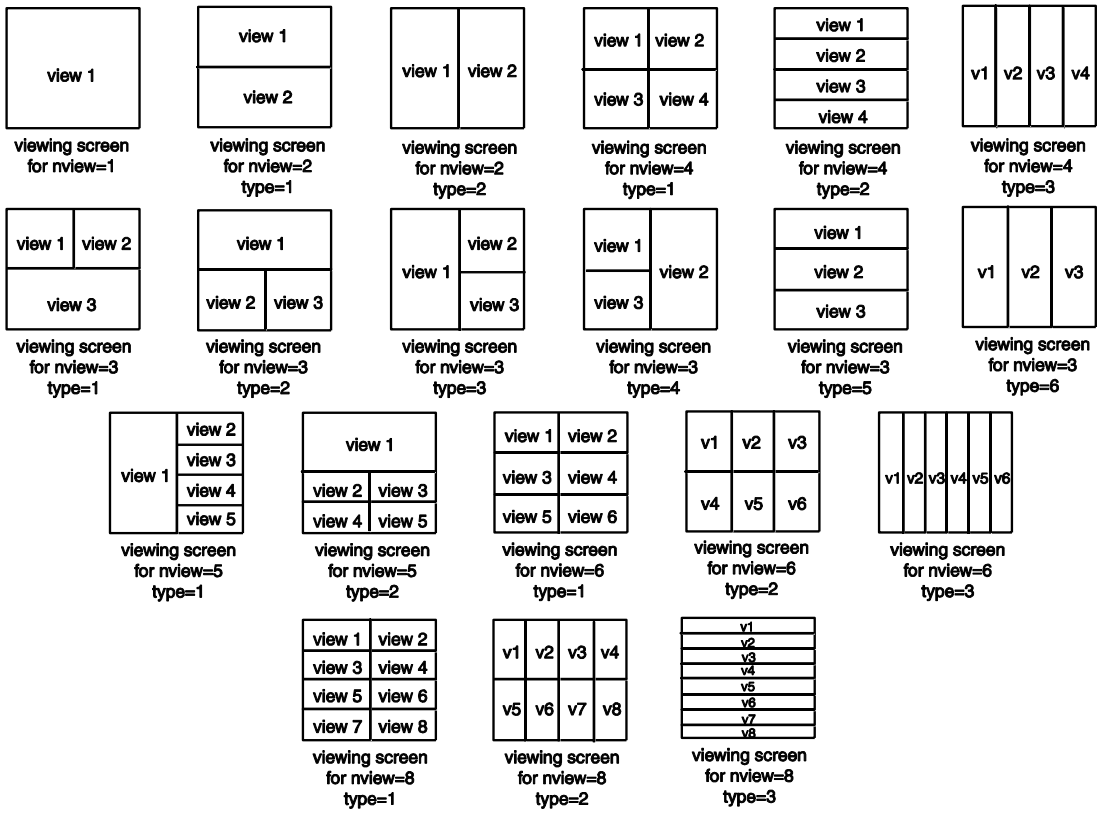
Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

NVEW *nview type*

nview = (I) Number of viewing windows.
Any of: 1, 2, 3, 4, 6 or 8
Default = 1

type = (I) For *nview*=2, 3, 5, 6 or 8, *type* specifies which type of multiview arrangement to use, as shown below.
Default = 1



PLOT (curves) Subcommand
GRPH (graphics) Input Group **GRPH-PLOT**

Purpose: To plot a curve or group of curves (up to 6 per plot).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

PLOT *[curve 1] [curve 2] . . . [curve 6]*

curve n = (X) Defines a single x-y curve to be plotted. This construct can have the following forms:

Form 1: *filename record*
Form 2: *record*
Form 3: *filename record* VS. *filename record*
Form 4: *filename record* VS. *record*
Form 5: *record* VS. *record*

filename = (C) Name of file containing the time history record

record = (I) Record number on file

Examples:	**PLOT** 1 2 3 4 5 6	cross-plot the first 6 time history records from the first file defined with a **FILE** command.
	PLOT F2 1 2 3 F4 1 2	cross-plot the first 3 time history records from file F2 and the first 2 time history records from file F4.
	PLOT F2 1 VS. F4 3	create an x-y plot in which the data values from record 1 of file F2 provide the x (horizontal) values and record 3 of file F4 provide the y (vertical) values. The time information for these records is ignored. There should be the same number of data points in both records.

PLOT (trace plots) Subcommand
GRPH (graphics) Input Group **GRPH-PLOT**

Purpose: To generate a trace plot, also referred to as "synthetic seismograms," "B-scans," or "waterfall plots." These are a group of time history or frequency domain traces with time (or frequency) as the x-axis. The y-axis origin of each data record is shifted to its spatial location. The traces are drawn in small strip windows within the plotting frame.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM SCOM input parameters

PLOT TRAC *filename space_coordinate [record specification]*

filename = (C) Name of file containing the time history records

space_coordinate= (C) Sny of: "x," "y," "z," or "none." If x,y, or z, that coordinate on file *filename* is used to place each curve in the proper location on the y-axis. If "none" is chosen, the curves are placed at spatial coordinates 1., 2., … in the order specified.

[record specification]= (C) Records to be plotted. Any of:
all - plots all records on the file
list 10 15 16 18 3 5… define a list of record numbers
sets 1 10 1 100 200 5 defines records via sets of
(begin, end, increment) triplets
dataname [ijk window] — requests all records of a particular name (e.g., pres) within the given ijk window. Here, the ijk window can be specified by "I," "j," or "k" followd by 1 or 2 values. E.g., pres j 10 identifies all pressure records having a j index = 10. xvel j 10 20 k 6 10 requests all xvel records within the range j=10 to 20, k=6 to 10.

Notes: 1. TRAC scales all curves to the same maximum (the peak of all displayed data)
2. See PSET/SET TRAC for control of magnification, size and clipping.

PLOT (plot data arrays) Subcommand
GRPH (graphics) Input Group **GRPH-PLOT**

Purpose: To plot information stored in a Data Manager data array.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
within **GRPH** Group: None.

PCOM SCOM input parameters

PLOT [*data1*] *(rangeoption minvalue maxvalue)*

[*data1*] = (X) Name of a data array stored within the Data Manager. If the array is in a directory, the path name should be given. For example: **PLOT** F1/DATA plots the time history data read to local file name F1 as a 2Dimensional color image. The array name is DATA and it is stored within directory F1.

rangeoption = (C) If blank, the color bar is autoscaled to the appropriate range of the requested data. If set to RANG, the user controls the color bar scale by using the following parameters.

minvalue = (C) Minimum data value to set the color scale.
Default = minimum data value contained in all *datan.*

maxvalue = (C) Maximum data value to set the color scale.
Default = maximum data value contained in all *datan.*

Note: If RANG and *minvalue /maxvalue* are input, the color scale is set appropriately. Any data values in the data array being displayed that fall outside the specified display range are assigned the nearest color, and the end points of the numerical scale plotted on the screen are adjusted accordingly. Numerical values associated with interior colors of the color scale are unaffected.

Examples:

```
PLOT F1/DATA                          /* plot material properties for the model read in as local
                                         filename F1

PLOT BPAT BAMP I 30 EX K 20 /*  plot an i and k slice of the x-component of electric field for
                                         the data read in as local filename F3

PLOT F2 MATR K 1 8 J 1 5 I 1 8        /* plot material properties for the subregion of the model
                                         indicated by the indices ranges shown for the data
                                         read in as local filename F2

BLOK B1 1 10 1 10 1 10                /* define a subregion, B1 of the model to plot
PLOT F3 AMPZ BLOK B1                  /* plot the z-component steady-state amplitude of the total
                                         electric field for the subregion defined by block B1 using
                                         data read in as local filename F3

PLOT F2 EZ J 6 K 8 VS. X              /* plot the z-component of the electric field along a single
                                         column of elements having j-index = 6, k-index = 8, and
                                         default i-indices ranging from 1 to the maximum i-index as
                                         an xy plot of value verses x-coordinate. The data used for
                                         plotting was read in as local filename F2.
```

PS (PostScript) Subcommand
GRPH (graphics) Input Group **GRPH-PS**

Purpose: To make PostScript image files for display on PostScript compatible devices such as laser printers.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: Must follow the graphics commands that generate the image.

PCOM	SCOM	input parameters
	PS	*option (filename)*

option = (C) Set to NOW, ON, OFF, or FILE. If set to NOW, creates a PostScript image of the current graphics display. If set to ON, creates a single PostScript image of any following PLOT commands until a PS OFF, subcommand is encountered.
If set to FILE a new root file name can be entered for a following set of images. The page numbering is reset to 1.
Default = NOW, unless a PS ON subcommand was input prior, in which case default = OFF.

Note: The **PS** subcommand writes the PostScript Image file for the current image displayed. All commands required to create the complete image must have been input previously. Any number of PostScript images may be produced during a job execution.

Example: **GRPH**

```
NVEW 2          /* sets number of views to 2
PLOT 1          /* plots first time history record in view 1
PLOT 2          /* plotssecond time history record in view 2
PS              /* creates PostScript file of the current  image
END
```

SET/PSET (set/preset) Subcommand
GRPH (graphics) Input Group **GRPH-SET/PSET**

Purpose: To set various graphics parameters. Syntax for the **SET** and **PSET** subcommands are the same. If applicable, the **SET** option sets graphics parameters for graphics in the current view. The **PSET** subcommand changes the code default for the parameter for all future plots. For options that cannot be changed once the plot is drawn, use of the **SET** command is interpreted as the **PSET** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: The action of the **PSET** subcommand applies to all subsequent plots. The action of the **SET** command applies only to the current plot.

PCOM SCOM input parameters

PSET *option parameters*
SET *option parameters*

option = (C) Option to set with this subcommand.

parameters = Number and types of input parameters vary depending on *option* as described below.

for option = TERM; lets the user turn off the graphics display on a terminal to allow PostScript graphics images to be created in batch mode and on systems that do not support X11 graphics display.

PSET TERM *switch*

switch = (C) Set *option* to X or OFF. Default is X. Use this option as the first input subcommand after the first occurrence of the **GRPH** command in order to deactivate the graphics terminal. If omitted or set to X, X11 graphics libraries must be present. If set to OFF, no graphics operations to the terminal screen are performed. This allows the generation of PostScript graphics in batch mode.

for option = FORM; allows the generation of PostScript images in either landscape or portrait form

PSET FORM *type*

type = (C) Set *type* to LAND (landscape) or PORT (portrait). Default is LAND. This option affects only the PostScript image that is generated. It does not affect the screen image.

for option = WNDO; causes a time history to be displayed using the user-provided horizontal and vertical data ranges. The **PSET** option applies to all future plots. The **SET** option applies to the current active view (either the last view displayed or the view set with the **VIEW** subcommand) unless the *alloption* parameter is specified. The window can be deactivated with the OFF option

PSET/SET WNDO *xbegin xend ybegin yend alloption*
or
PSET/SET WNDO *option*

xbegin, xend = (F) Beginning and ending range of the horizontal data window that the user wants to display. Typically, this is the time range for the x-y plot. Default is previous values.

ybegin, yend = (F) Beginning and ending range of the vertical data window that the user wants to display. Typically, this is the data range for the x-y plot. Default is previous values.

alloption = (C) If set to ALL, the data window specified is set for all views currently displayed.

option = (C) Set to OFF to deactivate window specification.

for option = DATE; controls whether or not the creation data and time are placed on the PostScript image.

PSET DATE *switch*

switch = (C) Set *switch* to ON or OFF. Default is ON. If ON, creation date and time are placed on the Postscript image.

for option = LW; controls the line width used to draw a PostScript image. The default line width is .75 points. This option applies to all views.

PSET LW *width*

width = (F) Width of lines in points where 1 point = 1/72 inch.

for option = LC; controls line color of curves drawn on an x-y plot. If a curve is given a line color it is also given a line quality of 1 (solid line). A user may assign an alternative line quality after the line color has been turned on.

PSET/SET LC *icurve icolor*
or
PSET/SET LC *option*

icurve = (F) Curve number (between 1 and 6)

icolor = (F) Color number (from the MATR color table currently defined) to assign to *icurve*

option = (C) Set to ON for all curves to have default colors 1 through 6
Set to OFF to turn line color off for all curves.

for option = LQ; controls the line quality to use for plotting x-y curves.

PSET/SET LQ *icurve linequal alloption*

icurve = (I) Curve number whose line quality is to be changed. This allows control over the line quality of each curve in a cross-plotted x-y view.

linequal = (I) Line quality to use (1= solid, 2= dashed, etc.). If positive value from 1 to 6, line quality is set to the specified value. If 0, line disappears from view. If negative value from -1 to -6, line quality is set to the absolute value of *linequal* and the curve legend disappears from the view.

alloption = (C) If set to ALL, then the line quality specified is set for all x-y type views currently displayed.

GRPH-SET/PSET

for option = XLAB; change the label of the horizontal axis for an x-y plot

PSET/SET XLAB *label alloption*

label = (C) Label to use, may be up to 80 characters. Enclose in single quotes (') if the label contains embedded blanks.

alloption = (C) If set to ALL, the x label specified is set for all views currently displayed.

for option = YLAB; change the label of the vertical axis for an x-y plot

PSET/SET YLAB *label alloption*

label = (C) Label to use, may be up to 80 characters. Enclose in single quotes (') if the label contains embedded blanks.

alloption = (C) If set to ALL, the y label specified is set for all views currently displayed.

for option = CLAB; change a curve label

PSET/SET CLAB *icurve label alloption*

icurve = (I) Curve number of the label to be changed. This allows control over the labeling of each curve in a cross-plotted x-y view.

label = (C) Label to use, may be up to 80 characters. Enclose in single quotes (') if *label* contains embedded blanks.
Note: Setting the last character of *label* to the plus (+) character causes *label* to be used in addition to (and preceding) the current or default curve label.

alloption = (C) If set to ALL, the curve label specified is set for the *icurve* curve for all views currently displayed.

for option = SWAP; swap the x and y axes for x-y plots

PSET/SET SWAP *switch*

switch = (C) Set *switch* to ON or OFF. Default is OFF. If ON, the x and y axes are swapped for the current view.

GRPH-SET/PSET

for option = STTL; defines a subtitle to be placed at the bottom of a view window the next time the view is drawn on the screen.

PSET/SET STTL *iview subtitle*

iview = (I) View number for which this subtittle will be used.

subtitle = (C) Subtitle to use, may be up to 80 characters. Enclose in single quotes (') if the subtitle contains embedded blanks.

for option = LOG; converts x-y graph scales from linear to logarithmic.

PSET/SET LOG *axis switch*

axis = (C) either X or Y.

switch = (C) either ON or OFF.

for option = DB; converts x-y graph scales from linear to dB scale according to the relation:

$$f'(x) = 20\log 10\left(\frac{f(x)}{dbvalue}\right).$$

PSET/SET DB *axis switch dbvalue*

axis = (C) either X or Y.

switch = (C) either ON or OFF.

dbvalue = (F) Value to use for computing dB. If not input, the max value for the curve in the current window is used.

for option = CORD; requests that default curve labels for x-y curve plots use xyz-coordinates instead of ijk-indices. This option must be preset using PSET. The SET option cannot be used to change the form of curve labels once a curve is already displayed. Coordinate information is available only for time histories of nodal or elemental field variables. If other types of curves are plotted with the CORD option set to ON, coordinates are displayed as 0.0

PSET CORD *switch*

switch = (C) either ON or OFF.

GRPH-SET/PSET

for option = TRAC; controls display of TRAC plots (see the **PLOT** TRAC subcommand). TRAC plots are sometimes refered to as "synthetic seismograms," "waterfall plots," or "B-scans."

SET/PSET TRAC *magnification strip_size clipping_boundary*

magnification = (F) Magnification. E.g., magnification = 2 magnifies data by a factor of 2 within its strip window.

strip_size = (F) Size of each strip window in spatial coordinate units.

clipping_boundary = (F) Defines where the data are clipped to the strip window. E.g., 1.0 clips data to the boundary of its own strip window, 2.0 clips well outside the strip window.

SIZE Subcommand
GRPH (graphics) Input Group **GRPH-SIZE**

Purpose: To allow the user to make PostScript x-y plots with a specified scale/inch This option does not affect the size of the image on a graphics terminal.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	SIZE	*xsize ysize xbegin xend ybegin yend nxdiv nydiv*

xsize = (F) Horizontal dimension of the graph window.
Set to OFF to deactivate preceding **SIZE** subcommand.

ysize = (F) Vertical dimension of the graph window.

xbegin, xend = (F) Beginning and ending values defining the horizontal data range.

ybegin, yend = (F) Beginning and ending values defining the vertical data range.

nxdiv = (F) Number of horizontal divisions on graph.

nydiv = (F) Number of vertical divisions on graph

Notes: 1. The subcommand:
SIZE OFF
causes future PostScript plots to be made using the codes auto scaling logic.
2. Only one set of size specifications can be specified at any one time. All views displayed are affected by the **SIZE** subcommand.

TTL (title) Subcommand
GRPH (graphics) Input Group **GRPH-TTL**

Purpose: To provide titling information for the graphic image currently displayed on the screen. The title is replaced without the image's being redrawn. Up to three lines of titling information can be placed on a plot.

Use Is: Optional. If no **TTL** subcommand is input,no title is displayed.

Multiple Use: Allowed.

Order Dependence
within **GRPH** Group: None.

PCOM SCOM input parameters

TTL *ititle* (FROM *localfile*)
<----------200 character title information for line = *ititle*---------->

ititle = (I) Title line number.
If ititle is not followed by FROM, the next input line (starting with column 1) contains up to 200 characters of text to substitute for title line = *ititle.* Up to three lines of titling information may be provided. Blank title lines are allowed if no title is desired.
(i.e., $1 \leq ititle \leq 3$).

Default = 1

localfile = (I) Optional paramerer. To use a title taken from one of the data files previously read into **Review** using the **READ** command, place the keyword FROM followed by the appropriate local file name on the **TTL** command line. In this case, the succeeding line of text is not input.

Default = none

Note: Once specified, the title information is used for all succeeding graphics until it is changed with the **TTL** subcommand.

VIEW Subcommand
GRPH (graphics) Input Group **GRPH-VIEW**

Purpose: To specify the next view window to plot within. View numbers are as indicated for the **NVEW** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **GRPH** Group: None.

PCOM	SCOM	input parameters
	VIEW	*iview*

iview = (I) Viewing window to use for the next plot .
Applicable only for *nview* > 1.
Default = 1

Note: If the **VIEW** subcommand is not input, the plot window cycles from 1 through *nview* and then back to 1 as each **PLOT** subcommand is encountered.

WRIT (write) Subcommand
GRPH (graphics) Input Group **GRPH-WRIT**

Purpose: To write out the x and y values of xy plots in the current view to a file as column data.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **GRPH** Group: None.

PCOM SCOM input parameters

WRIT (*options*) *filename*

options = (C) Optional input parameters. May be set to any or all of: ONEX, NOLB, FORM. If more than one curve is plotted in the current view, set to ONEX to have only the x-data values from the first curve written out with the y-data values from all curves. Otherwise, the x-data values from each curve are written in column form. If curve labels for each column of data are not desired, they may be suppressed with the NOLB (no label) option. The FORM option allows the user to change the default output format of the data. This option must be followed by a format specification. See Note 4 below.

filename = (C) File name to which the column data are to be written.

Notes: 1. The x-y data written to *filename* is for the current view. The current view is the last view plotted. The current view may be changed by using the **VIEW** subcommand.
2. The data written to *filename* are in column form. The first column contains the x-values of curve 1, the second column contains the y-values of curve 1, and the third column contains the x values of curve 2, etc. This assumes that the ONEX option was not specified.
3. The first line of *filename* contains character labels identifying the data in each column unless the NOLB option was specified.
4. Up to 12 columns may be written. The default FORTRAN format is (1p,12e12.4). The user may use the FORM option followed by a format to override the default format. For example: WRIT FORM (12f15.3) MYFILENAME

HIST (History) Input Command

Purpose: To manipulate time history files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM SCOM input parameters

HIST

COMB	*in1filename in2filename outfilename*
FFLD	*[histname] direction coord_side delcoord wavespeed density* & *timebegin timeend timeinc timeshift typehist outfile*
SMI	(complex command structure, see subcommand documentation)
END	

COMB (combine) Subcommand
HIST (history) Input Group **HIST-COMB**

Purpose: To combine two partial time history files (flxhst type) from succeeding restart runs into a single time history file.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **HIST** Group: None.

PCOM SCOM input parameters

COMB *in1filename in2filename outfilename*

in1filename = (C) Name of the first input time history file.
No default.

in2filename = (C) Name of the second input time history file. The time range for this file must follow the time range contained on *in1filename*.
No default.

outfilename = (C) Name of the combined time history file.
No default.

FFLD Subcommand
HIST (history) Input Group **HIST-FFLD**

Purpose: To generate a freefield (flxffl type) boundary file from a velocity or pressure-time history assuming a uniform plane wave, normally incident to the boundary. This boundary file can be applied as an FFLD boundary condition to the side of a **PZFlex** model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **HIST** Group: None.

PCOM SCOM input parameters

FFLD *[histname] direction coord_side delcoord wavespeed density* &
timebegin timeend timeinc timeshift typehist outfile

histname = (X) Specification of the time history that defines the time history of the incident wave. Can be any of:
(a) using the FUNC command to define the time history
(b) Name of the time history created using the **DATA HIST** command.
(c) Localfilename and record number of a time history read in using the READ command, i.e., F1 5
No default.

direction = (C) Direction the wave is traveling.
Any of: X+, X-, Y+, Y-, Z+, Z-
No default.

coord_side = (F) *Direction* coordinate of the side of the model to which the boundary is to be applied.
Default = 0.0

delcoord = (F) Element spacing normal to the boundary to which the FFLD boundary is to applied. No default.

wavespeed = (F) Normal wavespeed of the material at the boundary.
No default.

density = (F) Density of the material at the boundary.
No default.

timebegin = (F) Beginning time to be placed on the boundary file
Default = beginning time of *histname*

timeend = (F) Ending time to be placed on the boundary file
Default = ending time of *histname*

timeinc = (F) Time increment to be placed on the boundary file
Default = the time increment of *histname*

timeshift = (F) Time shift to add to the time history before creating the boundary file
Default = 0.0

typehist = (C) Type of time history. Either VEL (velocity) or PRES (pressure). Default = PRES.

fileout = (C) Name of the freefield boundary file created.
Default = fflout.

SMI (structure-media interaction) Subcommand
HIST (history) Input Group **HIST-SMI**

Purpose: To create complex FFLD boundary condition files from time histories generated by foreign codes (pecialized option).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **HIST** Group: None.

PCOM	SCOM	input parameters
	SMI	*nblocks timescale histfile fileout* *axistype ffc1 ffc2 ffc3 ffv1 ffv2 ffv3 ffs1 ffs2 ffs3 ffs4 ffs5 ffs6 ffp*

(For each data block, input the following)
datatype nffc1 nffc2 nffc3 lffc1 sffc1 lffc2 sffc2 lffc3 sffc3 sdata
irbeg1 irend1 irinc1 (repeat as needed on this line in sets of 3)
(input one line for each active component)
irbegn irendn irincn (repeat as needed on this line in sets of 3)

nblocks = (I) Number of data blocks to create. For one boundary, this is typically 2, a velocity and a stress block. No default.

timescale = (F) Scale factor to convert the time on *histfile* to **PZFlex** model time. Default = 1.

histfile = (C Local filename containing the time histories to use to construct the freefield boundary file.

fileout = (C) Name of the freefield boundary file created. No default.

axistype = (C) Type of freefield coordinate system. Any of: CART, CYLN, or SPHR. No default.

ffc1,..,ffc3 = (I) Freefield coordinate flags. Set to 1 if freefield data varies in along the 1, 2 or 3 direction. Otherwise, set to 0.
Default = 0.

ffv1,..,ffv3 = (I) Freefield velocity flags. Set to 1 if the velocity component is active in the freefield analysis. Otherwise, set to 0.
Default = 0.

ffs1,..,ffs6 = (I) Freefield stress flags. Set to 1 if the stress component is active in the freefield analysis. Otherwise, set to 0.
Default = 0.

ffp = (I) Freefield pressure flag. Set to 1 if the pressure component is active in the freefield analysis. Otherwise, set to 0.
Currently this should always be set to 0.
Default = 0.

datatype = (C) Any of: VEL, STRS or PRES. No default.

nffc1,..,nffc3 = (I) Number of freefield data stations in the 1, 2 and 3 coordinate directions. Must be at least 1. No default.

lffc1,..,lffc3 = (I) Position of the coordinates for the 3 coordinate directions in the coordinate data contained on *histfile* (1, 2 or 3). No default.

sffc1,..,sffc3 = (F) Scale factor to scale the coordinates contained on *histfile* to convert them to **PZFlex** model units. No default.

sdata = (F) Scale factor to convert the data records on *histfile* to **PZFlex** model units. No default.

irbegn,irendn,irincn = (F) Beginning, ending, and increment numbers defining a group of time history records on histfile that are to be mapped to the freefield data stations on the freefield boundary file. Define all the records for each active data component on a single line. The number of records must match the total number of freefield stations and must be mapped incrementing along coordinate direction 1, then 2, then 3. No default.

Notes:
1. For Cartesian models, direction 1 is x, 2 is y, and 3 is z.
2. For Cylindrical freefield models, direction 1 is radial, 2 is circumferential, and 3 is axial.
3. For spherical models, direction 1 is radial, 2 is theta, and 3 is phi. See the AXIS command documentation.

ID (identifier) Input Command

ID

Purpose: To set or change the identifier for a file.

Use Is: Optional. If not input, the file identifier is assumed to be the *id* parameter specified on the **TITL** command of the job that created the time history file.

Multiple Use: Allowed.

Order Dependence: Must follow the **READ** command that reads in the time history and identifier information for the file.

PCOM input parameters

ID *filename fileid*

filename = (C) Name of the file.

fileid = (C) An 8-character identifier for the file. No default.

Note: If time histories from different files are cross-plotted, *fileid* is added to each curve label to identify which file the curve belongs to.

INTG (Integrate) Input Command

Purpose: To integrate the specified time histories (i.e., compute displacement-time histories from velocity-time histories).

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the creation of the data file.

PCOM _____ input parameters _____

INTG *filename (label1 label2 . . . labeln)*

filename = (C) Name of the file containing time histories whose integral will be computed.

label1,. . .,
labeln = (C) Optional input parameters. Labels of data records to compute integral for (e.g., XVEL, SGYY, etc.) If no labels are input, all time histories are integrated. If labels are input, only data records with the labels specified are integrated.

Notes: 1. This command replaces the original data contained in the file with its integral. The original data are no longer available in *filename*.
2. If the user desires to maintain the original data in *filename* for plotting, the **MAKE** option provides an alternative way of creating new time histories that are the integrals of selected time histories contained in *filename*.

LIST Input Command

LIST

Purpose: To list the records contained in an internal file that was read in using the **READ** command. This option applies to listing the individual records contained in a time history file or listing "data out" snapshot arrays. For time histories, there is both a short and long form to the list. By default, the short form is provided. The list shows the position number of each record in the file as well as curve label and IJK indices. Use of the optional input parameters allows a subset of the records contained in the file to be listed. For "data out" files, the list of all saved data arrays and the times at which they were saved is provided.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the **READ** command that read in the data for a file.

PCOM input parameters

For listing time history files (see Note 1)

LIST *option*

option = (C) Select the list option for future LIST commands.
Either: SHORT or LONG
Default: SHORT

or

LIST *filename (label) (indxlabel index) (nbegin nend ninc)*

filename = (C) Name of the file to be listed.
No default.

label = (C) Optional parameter. If input, list only records with this curve label (e.g., XVEL, SGYY, etc.)
No default.

indxlabel = (C) The index label that denotes the type of *index*.
Any of: I, J, or K.
indxlabel must be immediately followed by *index*.
No default

index = (I) The index value of type *indxlabel*. No default.
List only records having the specified value of *index*.
Three sets of *(indxlabel index) parameters* may be specified, one for each index type.

nbegin, nend,ninc = (I) Beginning record, ending record, and increment to list from the file.

For listing "data out" snapshot files

LIST *filename nbegin nend ninc*

filename =	(C)	Name of the file to be listed. No default.
nbegin, nend,ninc =	(I)	Beginning time slice, ending time slice, and time slice increment to list from the file. The first time slice is identified at 1, the second time slice is 2, etc. Default: List all snapshot time slices contained on the file.

Note: If no list options are input when time history files are listed, all records on the file are listed. If one or more list options are input, only time history records that satisfy all input options are included in the list.

MAKE Input Command

MAKE

Purpose: To provide the capability to use mathematical operations to make new data records (typically time history or frequency domain records) by manipulating previously available records. This option provides the capability to create relative motion time histories, devide two frequency domain records, etc.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Should follow the commands that created the base set of time or frequency domain records that will be used to construct the new records.

PCOM SCOM _____ input parameters _____

MAKE

FILE *makefilename (option) fileid*

TTL
<----------60 character title for *makefilename* ---------->

TIME/FREQ FROM *filename*
or
TIME/FREQ *npoints valubegin valueinc*

CURV < mathematical expression defining new curve record >
CURV_ADD < mathematical expression defining curve to add to previous record >

DFT *filesignal irecsignal frequency option*

DHML *fileforc irecforc filestep irecstep*

ENVL *filesignal irecsignal*

IFFT *filesignal irecamp irecphas rectype*

LABL *label i-index j-index k-index x-coord y-coord z-coord*

POSVEC *axisname component*

SHIFT *filesignal irecsignal timeshift valueshift*

TFRM *axisname component option filename i-index j-index k-index*

WNDO *type option fraction windobegin windoend*

UNWRAP *filename record shifttype (ntstrt or rtstrt)*

Example 1: An example of a complete **Review** input file using the **MAKE** input command group to create 2 time histories that represent the relative velocity between two nodes whose velocity-time histories are contained on the disk file FLXHST.RESULTS and to save the results for future use.

```
READ    F1   FLXHST.RESULTS                  /* reads in base time histories
MAKE
        FILE     F2                  /* defines a new local file name for new records
        CURV     { F1  12 } - { F1  6 }      /* subtracts x-velocity records
        LABL     DELX                        /* labels new data record as DELX
        CURV     { F1  14 } - { F1  8 }      /* subtracts y-velocity records
        LABL     DELY                        /*labels new data record as DELY
        END
GRPH
        PLOT     F2  1  F2  2                /* plots relative velocity plots
        PS                                   /* creates PostScript image
        END
WRIT    F2  VELOCITY.DELTA                   /* write F2 to new disk file name
```

FILE Subcommand
MAKE Input Group

MAKE-FILE

Purpose: To define a new local file name that contains the new records created with **CURV**, **DHML,** and **ENVL** subcommands.

Use Is: Required.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must precede all other subcommands that define the new records to be placed in *makefilename*.

PCOM	SCOM	input parameters
	FILE	*makefilename (option) fileid*

makefilename = (C) Name of the new local file to contain new data records created by **CURV**, **DHML,** and **ENVL** subcommands.

option = (C) Optional input. Set to CLER to clear all records previously stored in local file *makefilename*. Otherwise ignore this input.

fileid = (C) Identifier (up to 8 characters) to be associated with the file *makefilename*. This can also be set or modified with the **ID** command. This identifier affects curve labels when curves with different identifiers are cross-plotted. The identifier is associated with the file if it is written to a disk file with the **WRIT** command. If not input, *fileid* defaults to that of the file providing the first time history used by the first **CURV** subcommand encountered.

Notes: 1. Each **CURV**, **DHML,** and **ENVL** subcommand that follows a **FILE** subcommand adds a single new data curve to *makefilename*. If a new **FILE** subcommand is encountered, all subsequent subcommands add a curve to the new *makefilename*.
2. Subsequent **FILE** subcommands with the same *makefilename* cause curves to be appended to the file. Previous curves are removed or overwritten unless the CLER option has been specified.

TTL (title) Subcommand
MAKE Input Group **MAKE-TTL**

Purpose: To provide a title for the current *makefilename*. This title is used only if *makefilename* is written to an external disk file with the **WRIT** command. Otherwise, the command can be ignored.

Use Is: Optional. If not input, the file title defaults to that of the file providing the first data record used by the first **CURV** subcommand encountered.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	TTL	<----------60 character title for *makefilename* ---------->

Note: The title line must be input immediately after the **TTL** subcommand line.

MAKE Input Group **MAKE-TIME/FREQ**
TIME/FREQ Subcommand

Purpose: To define the time or frequency sampling for the file *makefilename* created with the last **FILE** subcommand. The functionality of the **TIME** and **FREQ** (frequency) subcommands are identical within the program.

Use Is: Optional. If not input, the time/frequency information for *makefilename* defaults to that of the first **CURV** or other record creation subcommand encountered.

Multiple Use: Allowed. Only one use allowed for each **FILE** subcommand.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies and precede all **CURV, DHML,** and **ENVL** subcommands that add new records to *makefilename*.

PCOM	SCOM	input parameters
TIME		FROM *filename*
FREQ		FROM *filename*
	or	
TIME		*ntime timebegin timeinc*
FREQ		*nfreq freqbegin freqinc*

filename = (C) If the first input parameter is FROM, *filename* defines another local file from which time or frequency sampling information is to be taken.

<for manipulating time history records>

ntime = (I) Number of data-time pairs to represent the time histories to be added to *makefilename*.
timebegin = (F) Beginning time of the time histories.
timeinc = (F) Time increment between time sampling points. The ending time , *timeend*, of the time histories is:
timeend = *timebegin* + *timeinc* x (*ntime* - 1)

<for manipulating frequency domain records>

nfreq = (I) Number of data-freqeuncy pairs to represent the frequency domain records to be added to *makefilename*.
freqbegin = (F) Beginning frequency of the data record.
freqinc = (F) Frequency increment between record sampling points. The ending frequency, *freqend*, of a frequency domain record is:
freqend = *freqbegin* + *freqinc* x (*nfreq* - 1)

Note: It is usually possible to default the time/frequency information for a new file and skip the use of the **TIME/FREQ** subcommand to explicitly set this information.

CURV (curve) Subcommand
MAKE Input Group **MAKE-CURV**

Purpose: To create a new data curve and add it to the curves contained on *makefilename*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	CURV	< mathematical expression defining new curve >

A new data record, $r_{new}(x)$, is created based on a mathematical expression $f(r_1(x),r_2(x),...,r_n(x))$, which may contain any of the Symbol math operators. Each r_n is a data record defined by the construct:

{ *filename record* }

where the braces are required, all inputs are blank delimited and:

filename = (C) Local file name containing the record to use

record = (I) Record number on the file

Notes: 1. The data values defining each record, r_n, specified are initially interpolated to the x data sampling (usually time or frequency) of *makefilename*. This sampling can be controlled using the **TIME/FREQ** subcommand. Then, at each time sampling point i, the value of $r_{new}(x_i)$ is computed by evaluating the mathematical expression using the values $r_1(x_i)$, $r_2(x_i),...,r_n(x_i)$.

2 New data curves created using the **CURV** subcommand have default curve labels, ijk-indices, and coordinates from curve h_1. The **LABL** subcommand placed immediately after the **CURV** subcommand allows the user to specify a new 4-character label and any ijk-indices that the user chooses. When plotting these curves, the user can specify any desired curve labels using the **GRPH SET** subcommand.

3. With an extension to the basic option just described, TIME or FREQ (for frequency domain data) may be placed within the braces of the data record construct, i.e., { TIME , instead of a file name and record number. This substitutes the value of x_I (ususally time or frequency) at the appropriate location within the mathematical expression as described in Note 1 instead of $r_n(x_i)$. Consequently, a user could use this option to generate any new function, $r_{new}(x_i)$.

Notes: 4. Another option provided to define $r_n(x_i)$ is { FUNC } which substitues the value of the function, previously defined using the **FUNC** command, evaluated at x_i.

5. Another extension is the addition of the INTG (integrate) and DERV (derivative) operators to those provided by Symbol. If INTG is placed just prior to a time history construct, for example: INTG { F1 4 }, the integral of the specified curve is used instead of the curve itself. This integration is performed without modifying the base curve in file F1. The same holds true for the DERV option. Note that the integral or derivative is computed for the curve after it has been interpolated to the x_i sampling assigned to *makefilename*.
6. Records from any local file can be used in the mathematical expression, including those previously created in *makefilename*.

Example 1: Compute a new data record that is the difference between the first and third record on file F1:

```
CURV  { F1  3 } - { F1  1 }
```

Example 2: Compute the time history of the absolute magnitude of displacement of a node relative to its initial position using the x-, y-, and z-displacements contained in the first 3 records of file F1.

```
CURV  SQRT ( { F1  1 } ** 2. + { F1  2 } ** 2. + { F1  3 } ** 2. )
```

Example 3: Compute the radial velocity (in the y-z plane) of a node at the 30-degree location on a cylinder whose y- and z- velocities are contained in records 5 and 9, respectively, of file F6. As the expression will not fit on a single line, a continuation line is used by placing an "&" at the end of the line to be continued. A maximum of 70 items can be placed on a single input command, including all continuation lines:

```
CURV  COS ( 30. * 3.14156 / 180. ) * { F6  5 } + &
          SIN ( 30. * 3.14156 / 180. ) * { F6  9 }
```

alternatively, making use of Symbol variables:

```
SYMB  ANGLE = 30.
SYMB  COSVAL = COS ( $ANGLE * 3.14159 / 180. )
SYMB  SINVAL =  SIN ( $ANGLE * 3.14159 / 180. )
CURV  $COSVAL * { F6  5 } + $SINVAL * { F6  9 }
```

Example 4: Create a new curve that is the integral of curve 5 in file F2 scaled by 100.

```
CURV  100.0 * INTG { F2  5 }
```

Example 5: Create a sinusoid, with a frequency of 90 cycles/unit time and an amplitude of 12.

```
CURV  12.0 * SIN ( 90.* 2. * 3.14159 * { TIME } )
```

CURV_ADD (curve add) Subcommand
MAKE Input Group **MAKE-CURV_ADD**

Purpose: To add a user defined curve to the last curve currently contained on *makefilename*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	CURV_ADD	< mathematical expression defining new curve >

The last data record, $r_{last}(x)$, is modified by adding the result of the current mathematical expression $f(r_1(x),r_2(x),...,r_n(x))$, which may contain any of the Symbol math operators. The syntax and use of the mathematical expression provided are identical to those of the **CURV** subcommand.

Notes: 1. If a user has not defined any curve records prior to using the **CURV_ADD** subcommand, the defined curve is the first record on the current *Makefile*.

2 When combined with Symbol language constructs, this option is a simple way to sum together a number of time history records. For example, the following commands create a time history on *Makefile* F2 that contains a single record that is the sum of the first 100 time history records on *localfile* F1.

```
MAKE
        FILE F2
        DO LOOP1 N 1 100
                CURV_ADD { F1 $N } /* add each curve together
        END$ LOOP1
        END
GRPH
        PLOT F2 1 /* plot the combined time history
        END
```

DFT (discrete Fourier transform) Subcommand
MAKE Input Group **MAKE-DFT**

Purpose: To compute the Fourier Transform of a record at a requested frequency. This command adds 2 records to the current makefile.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MAKE** Group: Must follow the MAKE-FILE subcommand.

PCOM	SCOM	input parameters
	DFT	*filesignal irecsignal frequency option*

filesignal= (C) The local file name of the file which contains the signal.

irecsignal= (I) The record on local file *filesignal* which contains the signal.

frequency = (C) The frequency to use in computing the transform. Default = 0.

option = (C) Eitther of Real or Amp. Real causes Real and Imaginary part of the transform to be computed. Amp causes amplitude and phasse to be computed.

Notes: 1. This command displays the Fourier Integrals vs. time. The transform is the final value.

DHML (Duhamel) Subcommand
MAKE Input Group **MAKE-DHML**

Purpose: To create a new time history record using Duhamel convolution and add it to the curves contained on *makefilename*. For linear systems, the response to any forcing function can be obtained by convolving the response to a step function with the desired forcing function.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	DHML	*fileforc irecforc filestep irecstep*

fileforc = (C) Local file name of the file that contains the forcing function.

irecforc = (I) Record on local file *fileforc* that contains the forcing function.

filestep= (C) Local file name of the file that contains the response to a unit step function.

irecstep = (I) Record on local file *filestep* thatcontains the response to a unit step function.

Note: Each **DHML** subcommand that follows a **FILE** subcommand adds a single new data curve to *makefilename*.

ENVL (envelope) Subcommand
MAKE Input Group **MAKE-ENVL**

Purpose: To create a new time history record containing the envelope of a signal and add it to the curves contained on *makefilename*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM	SCOM	input parameters
	ENVL	*filesignal irecsignal*

filesignal= (C) Local file name of the file that contains the signal.

irecsignal= (I) Record on local file *filesignal* that contains the signal.

Notes: 1. Each **ENVL** subcommand that follows a **FILE** subcommand adds a single new data curve to *makefilename*.

2. The envelope is defined as $E(t)=\sqrt{x(t)^2+y(t)^2}$ where x(t) is the signal and y(t) is its Hilbert transform.
3. FFT techniques are used to process the data records in order to compute the envelope function. As an FFT assumes a periodic signal, the envelope function also is periodic. It is often necessary to increase the length of the time record by using the **TIME** subcommand and displaying only the original time range using the **GRPH SET WNDO** option. This avoids artificial periodic behavior within the time range of interest.

LABL (label) Subcommand
MAKE Input Group **MAKE-LABL**

Purpose: To define a 4-character data label and ijk-indices for the last curve generated by a **CURV**, **DHML,** or **ENVL** subcommands.

Use Is: Optional. Default is described in Note 2 of **CURV** subcommand.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **CURV** or other record generation subcommand that created the curve being labeled.

PCOM SCOM input parameters

LABL *label i-index j-index k-index x-coord y-coord z-coord*

label = (C) 4-character data label. No default.

i-index= (I) i-index associated with this curve.
Default = current value.

j-index= (I) j-index associated with this curve.
Default = current value.

k-index= (I) k-index associated with this curve.
Default = current value.

x-coord = (F) x-coordinate associated with this curve.
Default = current value.

y-coord = (F) y-coordinate associated with this curve.
Default = current value.

z-coord = (F) z-coordinate associated with this curve.
Default = current value.

IFFT (inverse FFT) Subcommand
MAKE Input Group **MAKE-IFFT**

Purpose: To synthesize a new time history record from a frequency domain representation and add it to the curves contained on *makefilename*.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM SCOM input parameters

IFFT *filesignal irecamp irecphas rectype*

filesignal= (C) Local file name of the file that contains the frequency domain information. See Note 1.

irecamp= (I) Record on local file *filesignal* that contains the amplitude or real part.

irecphas= (I) Record on local file *filesignal* that contains the phase or imaginary part.

rectype= (C) Type of frequency domain information. Enter *amp* for amplitude & phase or *real* for real and imaginary.
Default = *amp.*

Notes: 1. Each **IFFT** subcommand that follows a **FILE** subcommand adds a single new data curve to *makefilename*.
2. Phase must be in degrees.
3. FFT techniques are used to process the data records. *filesignal* must contain an equally spaced power of 2 number of data points. If it comes from the FREQ command in Review, this will be the case.

SHIFT Subcommand
MAKE Input Group **MAKE-SHIFT**

Purpose: Shift a curve in time and/or value. Does not change the reference curve, but adds a new curve with the requested shifts.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MAKE** Group: Must follow the MAKE-FILE subcommand.

PCOM SCOM input parameters

SHIFT *filesignal irecsignal timeshift valueshift*

filesignal= (C) The local file name of the file which contains the signal.

irecsignal= (I) The record on local file *filesignal* which contains the signal.

timeshift = (F) The timeshift. Default = 0.

valueshift = (X) The valueshift. If default, the curve will be shifted so that the first value is zero.

Notes: 1. This command is useful for rezeroing a curve.

POSVEC (position vector) Subcommand
MAKE Input Group **MAKE-POSVEC**

Purpose: To define a local coordinate position vector for mapping the time-history data array to. Also sets the Review data array "data" to type 7.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the MAKE-FILE subcommand.

PCOM SCOM input parameters

POSVEC *axisname component*

axisname = (C) Name of the previously defined local coordinate system. Default = global coordinate system.

component = (C) Specific component of the transformed coordinate tensor to use as the position vector for the time-history data array.
Any of: X,Y or Z
Default = X.

Note: This command accomplishes the following tasks:

- Sets data manager array "data" containing the time-histories in the current file to mapping type 7
- Transforms the global coordinates of the time-histories into the local system

Maps "data" to time along the abscissa and the specified local position component along the ordinate and creates the axis labels "time" and "position"

TFRM (transform) Subcommand
MAKE Input Group **MAKE-TFRM**

Purpose: To create a new time history by transforming the time histories defining all components of a vector (such as velocity or displacement) or a tensor (such as stress or strain) into the local system and for the specific component requested.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM SCOM input parameters

TFRM *axisname component option filename i-index j-index k-index*

axisname = (C) Name of the local coordinate system (defined by the **AXIS** command) that defines the system to transform to.

component = (C) Specific component of the transformed vector or tensor to compute.
For *option*=DISP or VEL, any of: X,Y or Z
For *option*=STRS or STRN, any of:XX, YY, ZZ, XY, YZ, XZ
For *option*=RGVL or RGFR, any of: X, Y, Z, RX, RY, or RZ

option = (C) Any of: VEL (velocity), DISP (displacement), STRS (stress) or STRN (strain)), RGVL (rigid velocity), or RGFR (rigid force).

filename = (C) Name of the local file that contains the data to be transformed.

i-index, j-index, k-index = (I) Nodal or elemental indices identifying the node or element data to transform. For RGVL and RGFR enter the substructure number for i-index and leave j-index and k-index blank.

Notes: 1. The program searches for time histories of all components of the vector or tensor requested to transform. If time histories of some components are not found, these components are assumed 0.0 when performing the transformation

UNWRAP Subcommand
MAKE Input Group **MAKE-UNWRAP**

Purpose: To unwrap discontinuous phase records to obtain a continuous curve in the frequency domain.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM SCOM input parameters

UNWRAP *filename record shifttype (ntstrt* or *rtstrt)*

filename = (C) Local file name containing the record to use. No Default.

record = (I) Record number on the file. No Default.

shifttype = (C) Either "rad" or "deg" to indicate whether phase curve is plotted in radians or degrees. Points are shifted by ± 2π for "rad" option and ±360 degrees for "deg" option. Can alternatively enter a numerical value for the phase shift magnitude. No Default.

ntstrt,rtstrt = (I,F) A starting point along the frequency axis that is plotted correctly, in relation to which all other points are determined. Either an integer indicating the timestep number or a floating point number indicating the time/ frequency on the abscissa should be entered. Default is ntstrt = 1

Notes:
1. This subcommand keeps the old record unchanged and creates a new record in the current *makefile*.
2. Selecting either *shifttype* = "rad" or *shifttype*= "deg" does not change the solution, only the appearance of the plot.
3. The logic for this subcommand unrolls in both directions from the starting value *ntstrt* and either leaves points as they are or shifts them by ±2π such that each point's value is as close to the preceding point as possible.

UNWRAP Subcommand
MAKE Input Group **MAKE-UNWRAP**

Purpose: To unwrap discontinuous phase records to obtain a continuous curve in the frequency domain.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence
Within **MAKE** Group: Must follow the **FILE** subcommand to which it applies.

PCOM SCOM input parameters

UNWRAP *filename record shifttype (ntstrt* or *rtstrt)*

filename = (C) The local file name containing the record to use. No Default.

record = (I) The record number on the file. No Default.

shifttype = (C) Either 'rad' or 'deg' to indicate whether phase curve is plotted in radians or degrees. Points are shifted by ± 2π for 'rad' option and ±360 degrees for 'deg' option. Can alternatively enter a numerical value for the phase shift magnitude. No Default.

ntstrt,rtstrt = (I,F) A starting point along the frequency axis that is plotted correctly, from which all other points will be determined in relation to. **Either** an integer indicating the timestep number **or** a floating point number indicating the time/ frequency on the abscissa should be entered. Default is ntstrt = 1

Notes: 1. This subcommand keeps the old record unchanged and creates a new record in the current *makefile*.
2. By selecting either *shifttype* = 'rad' or *shifttype*='deg', the solution is not changed, only the appearance of the plot.
3. The logic for this subcommand unrolls in both directions from the starting value *ntstrt* and either leaves points as they are or shifts them by ±2π such that each point's value is as close to the preceding point as possible.

WNDO (window) Subcommand
MAKE Input Group **MAKE-WNDO**

Purpose: To define a windowing function by which to scale a time history. This option allows any time history record to be zeroed at either end of the time history. The **WNDO** subcommand applies to all following **CURV** and **CURV_ADD** subcommands for this instance of the **MAKE** command only or until it is deactivated.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **FREQ** Group: Must precede any **CURV** or **CURV_ADD** subcommand for which the windowing function is desired.

PCOM SCOM input parameters

WNDO *type option fraction windobegin windoend*

type = (C) Type of window function to use. See illustration.
Any of: BART = Bartlett window
WELC= Welch window
HANN = Hann window
OFF = Deactivates windowing function
Default = BART

option = (C) Window option. See illustration.
Any of: FULL = Full window
LEFT= Left side of window
RIGH = Right side of window

Default = FULL

fraction = (F) Fraction of the data range defined by *windobegin* and *windoend* to apply the LEFT or RIGH window option to. *fraction* is a value between 0. and 1. See illustration under the **FREQ WNDO** command.
Default = 1.0 means the scaling function varies over the entire time range specified.

windobegin= (F) Value of the window scaling function ranges from 0.0 to 1.0 in the data range *windobegin* to *windoend.* The value of the scaling function outside this range is 0.0. *windobegin* defines the begining time value to which a nonzero scaling value can be applied.
Default = The starting time of the record.

windoend= (F) *windoend* defines the ending time to which a nonzero scaling value may be applied.
Default = The ending time of the record.

Notes: 1. The function of this command is to scale the data values of a time history by the values of the window function at each corresponding time.
2. These window functions are discussed in the text *Numerical Recipes in FORTRAN, the Art of Scientific Computing - Second Edition* by W. H. Press, S. A. Teukolsky W. T. Vettering and B. P Flannery, Cambridge University Press, 1986.

MEM (memory) Input Command

MEM

Purpose: To change the memory allocation available to the current job from its standard default value.

Use Is: Optional. If not input, the default memory allocation is made for the job.

Multiple Use: Not allowed.

Order Dependence:Must be first input command.

PCOM _____ input parameters _____

MEM *memory_numeric memory_character option*

memory_numeric = (I) Number of millions of numeric data words to be allocated in RAM memory for this job. The number of bytes allocated is *memory_numeric* * 4 * 10^6 bytes. No default.

memory_character = (I) Number of thousands of 20 character data entries to be allocated in cpu memory for this job. The number of bytes allocated is *memory_character* * 20 * 10^3 bytes. No default.

option = (C) In general, users may ignore this input.
Set to NOPR (no print) to suppress the memory allocation message that is sent to the terminal screen when memory is allocated at the start of a run.
Set to DEBG (debug) to display some debug diagnostics.

Note: 1 Default values for *memory_numeric* and *memory_character* can be established within the Review.defaults file.

PS (PostScript) Input Command

Purpose: To manipulate PostScript files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:None.

PCOM	SCOM	input parameters
PS		
	NCPY	*psfilename ncopies ipagenumber*
	SPLT	*psfilename*
	CNVR	*psfilename*
	END	

NCPY (number of copies) Subcommand
PS (PostScript) Input Group **PS-NCPY**

Purpose: To create a modified PostScript file that produces multiple copies when disposed to a PostScript display device. Optionally, a page number may be added to each page.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PS** Group: None.

PCOM SCOM input parameters

NCPY *psfilename ncopies ipagenumber*

psfilename = (C) Name of the original PostScript file.
No default.

ncopies = (I) Number of copies desired each time the new file is disposed to a PostScript printer.
Default= 1.

ipagenumber = (I) If positive, the page number to be printed at the bottom center of each page contained on the file. If 0 or blank, no change is made to the page number on *psfilename*.
If -1, any page number contained on p*sfilename* is removed.
Default=0.

Notes: 1. The file produced by this subcommand is named *psfilename*.out.
2. *psfilename* is not changed by this subcommand.

SPLT (split) Subcommand
PS (PostScript) Input Group **PS-SPLT**

Purpose: To provide the option of splitting a PostScript file containing multiple pages into multiple files, each file containing an individual page.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PS** Group: None.

PCOM	SCOM	input parameters
	SPLT	*psfilename*

psfilename = (C) Name of the original PostScript file.
No default.

Notes: 1. The files produced by this subcommand are named *psfilename*.1, psfilename.2, etc.
2. *psfilename* is not changed by this subcommand.

CNVR (convert) Subcommand
PS (PostScript) Input Group **PS-CNVR**

Purpose: To import a PostScript file into various PostScript editor programs such as Adobe Illustrato,r which support the Adobe Illustrator Document Format, PostScript files produced by **PZFlex** and **Review** should first be converted using this subcommand. For files with multiple pages, this command also splits the files similarly to the **SPLT** subcommand.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence within **PS** Group: None.

PCOM	SCOM	input parameters
	CNVR	*psfilename*

psfilename = (C) Name of the original PostScript file.
No default.

Notes: 1. The files produced by this subcommand are named *psfilename*.1, psfilename.2, etc.
2. *psfilename* is not changed by this subcommand.

READ Input Command

READ

Purpose: To define a local file name, associate it with an external file pathname and read in all data from the external file. This command can read in time history (flxhst) files, as well as standard snapshot (flxdato) files.

Use Is: Required to bring in data for manipulation or graphical display.

Multiple Use: Allowed.

Order Dependence: Must precede any other commands which reference the local file name.

PCOM input parameters

<< Reading time history data files >>

READ *filename (option (lskip)) pathname (nbegin1 nend1 ninc1, ... , nbeginn nendn nincn)*

filename =	(C)	Local file name, i.e., F1, F2, etc.
option =	(C)	Optional input to specify a special format for file *pathname*. If set to COLM, formatted column data is read from *pathname*. If set to MAZD, a MAZ tecplot snapshot file format is used to read *pathname*.
lskip =	(I)	Optional input to specify the number of lines on file *pathname* to skip. If input, lskip lines (which might contain unwanted header information) is skipped before reading the column data.
pathname =	(C)	*pathname* of the disk file to be assigned to *filename*. The disk file denoted by *pathname* is opened and its data read in and placed in *filename*. *pathname* may be up to 80 characters long.
nbeginn, nendn, nincn =	(I)	One of n sets of optional time history ranges that define a subset of records to be read from *pathname* and assigned to *filename*. The three values that define a range are the beginning and ending records and the increment to use in reading records from *pathname*. If *nendn* < *nbeginn*, *nincn* must be negative. The user may specify up to 15 sets of ranges. Duplication of records to be read is allowed. If no time history range information is specified, the complete set of all the records on *pathname* is read in. No defaults for *nbeginn* and *nendn*. The default for *nincn* is 1 or -1, whichever is appropriate.

<< Reading snapshot data files. Created with **DATA OUT** commands >>

READ *filename pathname (nbegin nend ninc)*

filename = (C) Local file name, i.e., F1, F2, etc.
No default.

pathname = (C) *pathname* of the disk file to be assigned to *filename*. The disk file denoted by *pathname* is opened and its data read in and placed in *filename*. *pathname* may be up to 80 characters long.
No default.

nbeginn, nendn, nincn = (I) One of n sets of optional snapshot ranges that define a subset of the snapshot times contained on *pathname* to be read in and assigned to *filename*. The three values that define a range are the beginning and ending snapshot times and the snapshot increment to use in reading sets of snapshot data from *pathname*. If no snapshot range information is specified, the complete set of all snapshot records on *pathname* is read in. Use the **LIST** command to show the data arrays and internal file directories used to store the data.
No defaults for *nbeginn* and *nendn*. The default for *nincn* is 1.

Example 1: Assign local file name F3 to the time history data contained in disk file /U/USER/RESULTS/FLXHST.RUN1 and read in all time history data in preparation for plotting

READ F1 /U/USER/RESULTS/FLXHST.RUN1

Example 2: Assign local file name F3 to the column data contained in disk file /U/USER/RESULTS/DATA1 and read in all the data in preparation for plotting

READ F3 COLM /U/USER/RESULTS/DATA1

Example 3: Assign local file name F4 to the data contained in disk file FLXDATO.RUN1. and read in all timeslices of data into subdirectories F4/1, F4/2, F4/3 ...

READ F4 FLXDATO.RUN1

Example 4: Assign local file name F5 to the data contained in disk file FLXDATO.RUN2. and read in timeslices of data 1, 3, 5, and 10,11,12. These are stored in subdirectories F5/1, F5/3, F5/5, F5/10, F5/11, and F5/12. This option is useful when the size of the FLXDATO file is larger than available RAM.

READ F5 FLXDATO.RUN2 1 5 2 10 12 1

SCAL (scale) Input Command

Purpose: To scale the values contained in a time history data file.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the creation of the data file.

PCOM _____ input parameters _____

SCAL *filename option scalvalue (label1 label2 . . . labeln)*

filename = (C) Name of the file to be scaled.

option = (C) Scaling option. Any of:
TIME=scale time record for time history files
FREQ=scale frequency record for frequency domain files
DATA=scale data records
CORD=scale coordinates

Default = DATA.

scalvalue = (F) Scale value to use. Default = 0.

label1,. . .,
labeln = (C) Optional input parameters. Labels of data records to scale (e.g., XVEL, SGYY, etc.). If no labels are input, all file records are scaled. If labels are input, only data records having the labels specified are scaled.

SHFT (shift) Input Command

Purpose: To shift the values contained in a time history data file.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the creation of the data file.

PCOM _____ input parameters _____

SHFT *filename option shiftvalue (label1 label2 . . . labeln)*

filename = (C) Name of file to be shifted.

option = (C) Shifting option.
Any of:
TIME= shift time record
DATA= shift data records
CORD= shift coordinates
Default = DATA.

shiftvalue = (F) Shift value to use. Default = 0.

label1,. . .,
labeln = (C) Optional input parameters. Labels of data or coordinate records to shift (e.g., XVEL, SGYY, XCRD etc.). If no labels are input, all file records are shifted. If labels are input, only data records having the labels specified are shifted.

STOP Input Command

STOP

Purpose: To terminate the processing of a job in an orderly manner.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: None

PCOM	input parameters
STOP	-none-

Notes: 1. If a **STOP** command is encountered in the middle of the input stream, no input commands following the **STOP** command is processed.

2. If no **STOP** command is found and the end-of-file for the Job Input File is encountered, a **STOP** command is automatically generated resulting in a normal termination of job processing.

TERM (terminal) Input Command

Purpose: To allow the user to control the routing of standard input and output files to the terminal or to the Job Input and Job Output Files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: None.

PCOM _____ input parameters _____

TERM *inoption outoption*

or

T *inoption outoption*

inoption = (C) Option controlling standard input. Either ON or OFF.
If ON, the user must provide input commands from the terminal keyboard in interactive mode.
If OFF, input commands are read from the Job Input File.
Default: toggle from current state. If currently ON, default is OFF. If currently OFF, default is ON.

outoption = (C) Option controlling standard output . Either ON or OFF.
If ON, standard output from the code is written to the user's terminal screen.
If OFF, standard output is written to the Job Output File.
Default = *inoption*.

Notes:
1. Standard output displayed on a user's terminal is not recorded in the Job Output File.
2. If a user switches the standard input from the Job Input File to the terminal keyboard and later switches input back to the Job Input File, processing of commands from the Job Input File continues where it had left off.

Examples:

TERM /* toggle ON to OFF or OFF to ON

TERM OFF ON /* input is read from Job Input File, output is displayed at terminal

TERM ON OFF /* input from keyboard, output to Job Output File

WRIT (write) Input Command

Purpose: To write a group of time histories assigned to a file to the disk file specified.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:Must follow the commands that created the data on the local file name.

PCOM input parameters

WRIT *filename (option) pathname (nbegin1 nend1 ninc1, ... , nbeginn nendn nincn)*

filename = (C) Local file name, i.e., F1, F2, etc.

option = (C) Optional input to specify a special format for file *pathname*. If set to FRMT, then formatted data is written to *pathname*.

pathname = (C) *pathname* of the disk file to be written.
pathname may be up to 80 characters long.

nbeginn, nendn, nincn = (I) One of n sets of optional time history ranges that define a subset of records in filename to be written to *pathname*. The three values that define a range are the beginning and ending records and the increment to use in reading records from *pathname*. If *nendn* < *nbeginn*, *nincn* must be negative. The user may specify up to 15 sets of ranges. Duplication of records to be read is allowed. If no time history range information is specified, the complete set of all the records on *filename* is written.
No defaults for *nbeginn* and *nendn*. The default for *nincn* is 1 or -1, which ever is appropriate.

Note: Continuation lines cannot be used with the **WRIT** command due to parser constraints. If this is necessary, first use **MAKE** to collect the desired records into a single file before writing them out.

Example : Write all time histories in local file name F3 to the disk file /U/USER/RESULTS/FLXHST.A

```
WRIT  F3   /U/USER/RESULTS/FLXHST.A
```

4 BUILD - A GRID GENERATOR

Build is a utility program that simplifies the construction of skewed grids for **PZFlex**. The user interface for Build is similar to that of PZFlex (i.e., free format input, the use of the primary command/subcommand input structure, the ability to use SYMBOL commands anywhere in the Job Input File, etc). The availability of SYMBOL makes **Build** a powerful tool for grid generation, as input command files can be developed in a general form using variables as input parameters instead of specifying actual values. This allows for easy modification of models if desired.

A **Build** job requires a Job Input File and produces, if requested, a Grid Output File that can be used as a the Grid Input File for a **PZFlex** job. This file provides all the nodal coordinates for the skewed partition. If SYMBOL variables are used to define the grid geometry in **Build**, users usually find it convenient to create a Symbol File with the **SYMB** #SAVE option. This file can then be read by **PZFlex**, using the **SYMB** #READ command, in order to exchange all symbol values between **Build** and **PZFlex**.

This section briefly describes the **Build** commands and their use. For most commands, there are two specifications of the command input parameters. The first defines 2D models with x- and y-coordinates and i and j indices. The second defines the 3D form with x-, y-, and z-coordinates and i, j, and k indices. The 2D command form is simply a degenerative form of the 3D command with all z-coordinate or k-index input parameters ignored. They are written out separately for clarity only. Each **Build** command can be used as many times and in any order required. The following description of each command below discusses the full 3D form of the command.

LIST OF **BUILD** COMMANDS

PCOM	SCOM	input parameters
AXIS		
	DEFN	*axisname axistype x0 y0 z0 x1 y1 z1 x2 y2 z2*
	FORM	*option*
CORD		*i j x y icomp*
CORD		*i j k x y z icomp*
COPY		*ibf ief iif jbf jef jif iaxf icomp* *ibt iet iit jbt jet jit iaxt*
COPY		*ibf ief iif jbf jef jif kbf kef kif iaxf icomp* *ibt iet iit jbt jet jit kbt ket kit iaxt*
EQUV		*ifrom jfrom ito jto icomp*
EQUV		*ifrom jfrom kfrom ito jto kto icomp*
FILL		*(axisname) fillindex ibegin iend jbegin jend icomp*
FILL		*(axisname) fillindex ibegin iend jbegin jend kbegin kend icomp*
GEOM		/* for 2D or 3D models
	WNDO	*ibegin iend jbegin jend kbegin kend*
	XCRD	*xbegin xend ibegin iend ratio*
	YCRD	*ybegin yend jbegin jend ratio*
	ZCRD	*zbegin zend kbegin kend ratio*
GRID		*igrid jgrid*
GRID		*igrid jgrid kgrid*
LOCK		*option ibegin iend jbegin jend*
LOCK		*option ibegin iend jbegin jend kbegin kend*
LVAL		*i j ncomp value x1 y1 x2 y2 icomp*
LVAL		*i j k ncomp value x1 y1 z1 x2 y2 z2 icomp*
PRNT		*dataname option idirec ibegin iend jbegin jend*
PRNT		*dataname option idirec ibegin iend jbegin jend kbegin kend*
QUAD		*(axisname) ibegin iend jbegin jend icomp*
QUAD		*(axisname) ibegin iend jbegin jend kbegin kend icomp*
HEX		*(axisname) ibegin iend jbegin jend kbegin kend icomp*
RADI		*axisname radius ibegin iend jbegin jend*
RADI		*axisname radius ibegin iend jbegin jend kbegin kend*
READ		*form ibegin iend jbegin jend*
READ		*form ibegin iend jbegin jend kbegin kend*

PCOM SCOM input parameters

RFIL	*fillindex ibegin iend iinc jbegin jend jinc icomp*
RFIL	*fillindex ibegin iend iinc jbegin jend jinc kbegin kend kinc icomp*
ROT	*[axisname] delangle ibegin iend jbegin jend*
ROT	*[axisname] delangle ibegin iend jbegin jend kbegin kend*
RPOS	*i1 j1 i2 j2 i3 j3 i4 j4 icomp*
RPOS	*i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4 icomp*
SCAL	*xscale yscale ibegin iend jbegin jend*
SCAL	*xscale yscale zscale ibegin iend jbegin jend kbegin kend*
SET	*dataname value ibegin iend jbegin jend*
SET	*dataname value ibegin iend jbegin jend kbegin kend*
SHFT	*xshift yshift ibegin iend jbegin jend*
SHFT	*xshift yshift zshift ibegin iend jbegin jend kbegin kend*
SOR	*[axisname] circindex delangle ibegin iend jbegin jend*
SOR	*[axisname] circindex delangle ibegin iend jbegin jend kbegin kend*
STOP	
SVAL	*[axisname] value ncomp ibegin iend jbegin jend*
SVAL	*[axisname] value ncomp ibegin iend jbegin jend kbegin kend*
SWTH	*iaxes ibegin iend jbegin jend*
SWTH	*iaxes ibegin iend jbegin jend kbegin kend*
TERM	*inoption outoption*
TITL	*<--id--> <----------title----------->*
WRIT	*form ibegin iend jbegin jend*
WRIT	*form ibegin iend jbegin jend kbegin kend*

AXIS Input Command

AXIS

Purpose: To define local coordinate systems relative to the global Cartesian system. These new local systems can be accessed by other code options to facilitate transforms between global and local systems. The **AXIS** command group is identical to that described in Section 3. Refer to that section for a description of the options and parameters.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any other command that references the axis definitions.

PCOM	SCOM	input parameters
AXIS		
	DEFN	*axisname axistype x0 y0 z0 x1 y1 z1 x2 y2 z2*
	FORM	*option*
	END	

Notes:
1. The subcommands in the **AXIS** command group are identical to the equivalent subcommands of the **PZFlex** program. See Section 3 for details.
2. The **AXIS** command group defines one or more local coordinate systems. These coordinate systems may be Cartesian, cylindrical, or spherical. Its use and parameter description is identical to that described in Section 3. The **AXIS** command can be used in conjunction with various Build commands in order to provide automatic conversion from a local coordinate system to the global Cartesian coordinate system.

CORD (coordinate) Input Command

Purpose: Set the spatial coordinates of a single node in the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any other command that references the coordinates of the node.

PCOM	SCOM	input parameters
		<for 2D models>
CORD		*i j x y* (*icomp*)
		<for 3D models>
CORD		*i j k x y z* (*icomp*)

i, j, k = (I) I-, j- and k-indices for the node whose coordinates are being set.
No defaults.

x, y, z = (F) Values to set the node's x-, y-, and z-coordinates to.
Defaults = 0.0

icomp = (C) Optional parameter. Used to set the value of a single component of the node's coordinates while leaving the other two coordinate components unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

Examples.

Set all coordinates of the node having ijk-indices of (1,5,10) to the xyz coordinates (21.3, 30.0, .0021)

```
CORD  1  5  10  21.3  30.0  .0021.
```

Revise the coordinates of node (1,5,10) to be (21.3, 18.5, .0021) using the icomp parameter:

```
CORD  1  5  10  0.0  18.5  0.0  Y
```

COPY Input Command

COPY

Purpose: To copy the spatial coordinates of all nodes that fall within the "from" IJK window to the coordinates of all nodes that fall within the "to" IJK window. Both the "from" and "to" windows should be of matching size. Note that the copy command can use an increment. At the end of this command's action, the coordinates of nodes within both the "from" and "to" windows are identical.

Note: this is a 2-line command.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: The coordinates for nodes within the "from" window should be set prior to using this command. This command should precede any other commands that reference the coordinates within the "to" window.

PCOM SCOM input parameters

<for 2D models>

COPY *ibf ief iif jbf jef jif iaxf* (*icomp*)
ibt iet iit jbt jet jit iaxt

<for 3D models>

COPY *ibf ief iif jbf jef jif kbf kef kif iaxf* (*icomp*)
ibt iet iit jbt jet jit kbt ket kit iaxt

ibf, ief, iif = (I) Beginning I-index, ending I-index, and I-index increment for the "from" window. No defaults.

jbf, jef, jif = (I) Beginning J-index, ending J-index, and J-index increment for the "from" window. No defaults.

kbf, kef, kif = (I) Beginning K-index, ending K-index, and K-index increment for the "from" window. No defaults.

iaxf = (C) 3-character input that is any permutation of the characters: I, j, and k that defines the permutation of the "from" windows ijk-indices during the copy process.
For example, IJK, JKI, etc.
This input is of use only if the "from" and "to" windows are oriented differently in ijk space. See Note 1 below.
Default=IJK.

icomp = (C) Optional parameter. Used to copy the value of a single component of the node's coordinates while leaving the other two coordinate components unchanged.
Any of: X, Y, or Z. Default: all coordinate components are set with this command.

ibt, iet, iit = (I) Beginning I-index, ending I-index, and I-index increment for the "to" window. No defaults.

jbt, jet, jit = (I) Beginning J-index, ending J-index, and J-index increment for the "to" window. No defaults.

kbt, ket, kit = (I) Beginning K-index, ending K-index, and K-index increment for the "to" window. No defaults.

iaxt = (C) 3-character input that is any permutation of the characters: I, j, and k that defines the permutation of the "to" windows ijk-indices during the copy process.
For example, IJK, JKI, etc.
This input is of use only if the "from" and "to" windows are oriented differently in ijk space. See Note 1 below.
Default=IJK.

Note 1. The *iaxf* and *iaxt* parameters provide the functionality to map the I-index, J-index, and K-index of the "from" window to an alternative indices orientation in the 'to' window.
For example, if *iaxf* = IJK and *iaxt* = KJI, then the i-index range of the "from" window maps to the k-index range of the "to" window. Likewise, the k-index range of the "from" window maps to i-index range of the "to" window. The j-index range of both windows is the same.

Examples.

Copy all nodes in the window having i-indices in the range 11 to 20, j-indices in the range 6 to 8, and k-indices in the range 51 to 55 to another region of the grid with nodes having i-indices in the range 61 to 70, j-indices in the range 21 to 23, and k-indices in the range 96 to 100.

```
COPY    11   20   1    6    8    1    51   55   1
        61   70   1    21   23   1    96   100  1
```

Repeat this copy but copy only the z-coordinates. Also, map the "from" window indices (in the order IJK) to the "to" window indices with the order KJI. In this case, the "to" window is assumed to be defined as nodes having i-indices in the range 61 to 65, j-indices in the range 21 to 23, and k-indices in the range 91 to 100.

```
COPY    11   20   1    6    8    1    51   55   1    IJK  Z
        61   65   1    21   23   1    91   100  1    KJI
```

EQUV (equivalence) Input Command

EQUV

Purpose: To equivalence the coordinates of a single node to those of another node in the model. This function is similar to the **COPY** command except that it applies only to a single node in the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any other command that references the coordinates of the node whose coordinates are being assigned with this statement.

PCOM	SCOM	input parameters
		<for 2D models>
EQUV		*ifrom jfrom ito jto* (*icomp*)
		<for 3D models>
EQUV		*ifrom jfrom kfrom ito jto kto* (*icomp*)

ifrom, jfrom, kfrom = (I) i-, j-, and k-indices for the node whose coordinates will be used to set the coordinates of the "to" node.
No defaults.

ito, jto, kto = (I) i-, j-, and k-indices for the node whose coordinates will be set equal to those of the "from" node.
No defaults.

icomp = (C) Optional parameter. Used to equivalence only a single component of the "to" node's coordinates while leaving the other two coordinate components unchanged.
Any of: X, Y, or Z. Default: all coordinate components are set with this command.

Examples.

Set all three coordinate components of the node with ijk-indices (3,12,20) to those of the node having ijk-indices (1,1,1)

EQUV 1 1 1 3 12 20

Purpose: To compute the nodal coordinates for nodes in the interior of a bounded region by interpolation based on the coordinates of the bounding nodes of the region.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Must precede any other command that references the coordinates of the nodes assigned with this statement.

PCOM SCOM input parameters

<for 2D models>

FILL (*axisname*) *fillindex ibegin iend jbegin jend icomp*

<for 3D models>

FILL (*axisname*) *fillindex ibegin iend jbegin jend kbegin kend icomp*

axisname = (C) Optional input parameter. The name of a local axis system previously defined by the **AXIS** command group. This option is used only to perform the fill operation in a cylindrical or spherical coordinate system causing the interpolation to work directly with the curvilinear coordinates of the local system. This produces filled grid lines that are curved instead of straight. If not input, the fill operation works in the global Cartesian system.

fillindex = (C) The grid index fill parameter.
If a single index, any of: I, J, or K, this defines the index direction along which to fill.
For example, if *fillindex*=I, the coordinates of all nodes within the ijk window having *ibegin*< I-index < *iend* are defined by using equal-spaced interpolation based on the coordinates of the bounding nodes with
I-index=*ibegin* and I-index=*iend*.
If a double index, any combination of I, j, and k, for example JK, the program fills in the direction of the first index specified (J) but uses the coordinate positions on the bounding edges for the second index (K) in order to define the spacing along the first index. This allows for filling with non-uniform spacing. This option requires that the coordinates for the bounding edges of the two fill indexes be defined prior to use of this command.
No default.

ibegin, iend = (I) Beginning and ending I-indices of the bounding nodes for the region of nodes to be filled.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the bounding nodes for the region of nodes to be filled.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of the bounding nodes for the region of nodes to be filled.
Default: *kbegin* = 1, *kend* = maximum K

icomp = (C) Optional parameter. Used to perform the fill operation on only a single component of the coordinates for the nodes in the fill region. The other two coordinate components are left unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

Purpose: To generate a Cartesian mesh within an ijk-region (defined by the **WNDO** subcommand) of the mesh. This is similar to the standard partition geometry defined directly within **PZFlex** using the **GEOM** command group.

Use Is: Optional.

Multiple Use: Allowed. See Note 2 below.

Order Dependence: Must precede any other command that references the nodal coordinates generated by this command group.

PCOM	SCOM	input parameters
GEOM		
	WNDO	*ibegin iend jbegin jend kbegin kend*
	XCRD	*xbegin xend ibegin iend ratio*
	YCRD	*ybegin yend jbegin jend ratio*
	ZCRD	*zbegin zend kbegin kend ratio*
	END	

Notes:
1. The **GEOM** command group generates the coordinates for the nodes within the ijk-region defined by the **WNDO** subcommand *at the time* the **END** subcommand is encountered.
2. A separate **GEOM** command group must be used for each distinct Cartesian region (requiring a different **WNDO** subcommand) the user wishes to generate.
3. Although the coordinates of nodes generated using the **GEOM** command group are similar in form to those of a "standard partition" grid, they are treated the same as any other part of the skewed partition within the model.

WNDO (window) Subcommand
GEOM (geometry) Input Group **GEOM-WNDO**

Purpose: To define an ijk-region (window) within the model for which the set of **XCRD**, **YCRD,** and **ZCRD** subcommands define the descritization in the x-, y-, and z-directions. These coordinate directions map to the i-, j-, and k-indices directions of the grid, respectively.

Use Is: Optional. If not set, coordinates of the entire model are defined using this **GEOM** command group once the **END** subcommand is encountered.

Multiple Use: This command should be used only once in each **GEOM** command group.

Order Dependence within **GEOM** Group: None.

PCOM	SCOM	input parameters
	WNDO	*iwbegin iwend jwbegin jwend kwbegin kwend*

iwbegin, iwend = (I) Beginning and ending I-indices of nodes bounding the grid window whose coordinates are defined by this **GEOM** command group.
Default: *iwbegin* = 1, *iwend* = maximum I

jwbegin, jwend = (I) Beginning and ending J-indices of nodes bounding the grid window whose coordinates are defined by this **GEOM** command group.
Default: *jwbegin* = 1, *jwend* = maximum J

kwbegin, kwend = (I) Beginning and ending K-indices of nodes bounding the grid window whose coordinates are defined by this **GEOM** command group. Input only for 3D models.
Default: *kwbegin* = 1, *kwend* = maximum K

Purpose: To define nodal x-coordinates as a function of a node's I-index for nodes that fall within the window defined by the **WNDO** subcommand.

Use Is: Required. The x-coordinates of any nodes within the window that are not defined with **XCRD** subcommands are set to 0.0 once the **END** subcommand is encountered.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: The last **XCRD** subcommand that defines the x-coordinate of nodes with a specific i-index will control.

PCOM	SCOM	input parameters
	XCRD	*xbegin xend ibegin iend ratio*

xbegin, xend = (F) Beginning and ending x-coordinates bounding a region of the grid within which uniform discretization exist.
No defaults.

ibegin, iend = (I) Beginning and ending nodal I-indices of the grid section defined by *xbegin* and *xend*.
Default: *ibegin* = 1, *iend* = maximum I

ratio = (F) Size ratio of succeeding elements in the x-direction for elements of the the grid section bounded by *xbegin* and *xend*.
Default = 1.0

Note: Any x-coordinate specifications for nodes with I-indices that are outside the grid region defined by the **WNDO** subcommand are ignored.

YCRD (Y-coordinate) Subcommand
GEOM (geometry) Input Group **GEOM-YCRD**

Purpose: To define nodal y-coordinates as a function of a node's j-index for nodes that fall within the window defined by the **WNDO** subcommand.

Use Is: Required. The y-coordinates of any nodes within the window that are not defined with **YCRD** subcommands are set to 0.0 once the **END** subcommand is encountered.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: The last **YCRD** subcommand that defines the y-coordinate of nodes with a specific j-index will control.

PCOM	SCOM	input parameters
	YCRD	*ybegin yend jbegin jend ratio*

ybegin, yend = (F) Beginning and ending y-coordinates bounding a region of the grid within which uniform discretization exist.
No defaults.

jbegin, jend = (I) Beginning and ending nodal J-indices of the grid section defined by *ybegin* and *yend*.
Default: *jbegin* = 1, *jend* = maximum I

ratio = (F) Size ratio of succeeding elements in the y-direction for elements of the the grid section bounded by *ybegin* and *yend*.
Default = 1.0

Note: Any y-coordinate specifications for nodes with J-indices outside the grid region defined by the **WNDO** subcommand are ignored.

Purpose: To define nodal z-coordinates as a function of a node's k-index for nodes that fall within the window defined by the **WNDO** subcommand. Input only for 3D models.

Use Is: Required for 3D models. The z-coordinates of any nodes within the window that are not defined with **ZCRD** subcommands are set to 0.0 once the **END** subcommand is encountered.

Multiple Use: Allowed.

Order Dependence within **GEOM** Group: The last **ZCRD** subcommand that defines the z-coordinate of nodes with a specific k-index will control.

PCOM	SCOM	input parameters
	ZCRD	*zbegin zend kbegin kend ratio*

zbegin, zend = (F) Beginning and ending z-coordinates bounding a region of the grid within which uniform discretization exist.
No defaults.

kbegin, kend = (I) Beginning and ending nodal K-indices of the grid section defined by *zbegin* and *zend*.
Default: *kbegin* = 1, *kend* = maximum K

ratio = (F) Size ratio of succeeding elements in the z-direction for elements of the the grid section bounded by *zbegin* and *zend*.
Default = 1.0

Note: Any z-coordinate specifications for nodes with K-indices outside the grid region defined by the **WNDO** subcommand are ignored.

GRID Input Command

GRID

Purpose: To define the size of the mesh for which **Build** is going to define nodal coordinates. This should be the same size as the skewed partition of a **PZFlex** model.

Use Is: Required.

Multiple Use: Not allowed.

Order Dependence:Must precede all coordinate generation commands.

PCOM	input parameters
GRID	*igrid jgrid (kgrid)*

igrid = (I) Range of I-indices for the grid. No default.

jgrid = (I) Range of J-indices for the grid. No default.

kgrid = (I) Optional input. Range of K-indices for the grid. Input only for 3D models. No default.

Notes: 1. The ijk-indices of nodes whose coordinates are defined by **Build** are always assumed to range from 1 to *igrid*, 1 to *jgrid,* and 1 to *kgrid* in the I-, J-, and K-directions of the mesh. Coordinates for these nodes (or some subset of nodes) are written to the bldgrdo (Build grid output) file using the **WRIT** command. When these coordinates are read into the **PZFlex** program using the **GEOM SKEW** command, the nodal indices for these coordinates are revised to match the beginning and ending range of the skewed partition's I-, J-, and K-indices.

2. If *kgrid* is input, the grid generated is assumed to be 3D and all following commands are interpreted as such. If *kgrid* is not input, the model is assumed to be 2D and the 2D form of each command is assumed.

Examples: The following examples illustrate the use of the **GRID** command.

GRID 30 60 (defines a 2D mesh with 30 x 60 = 1800 grid nodes)
GRID 400 200 100 (defines a 3D mesh with 400 x 200 x 100 = 8,000,000 nodes)

Purpose: To lock or unlock the coordinates of nodes within the grid. Any attempt to change the coordinates of a node while locked results in a job error. Any nodes may be locked or unlocked at any point within the command input stream. This options guards against coordinates inadvertenly being changed by commands that follow the point at which the lock is applied.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence:All subsequent commands are subject to the constraints applied by the **LOCK** command.

PCOM	input parameters
	<for 2D models>
LOCK	*option ibegin iend jbegin jend*
	<for 3D models>
LOCK	*option ibegin iend jbegin jend kbegin kend*

option = (C) Lock option. Any of: ON or OFF. ON means nodes are locked against future modifications. OFF means that any current locks are removed.
No default.

ibegin, iend = (I) Beginning and ending I-indices of nodes bounding the grid region whose coordinates are locked or unlocked by this command.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of nodes bounding the grid region whose coordinates are locked or unlocked by this command.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of nodes bounding the grid region whose coordinates are locked or unlocked by this command.
Default: *kbegin* = 1, *kend* = maximum K

LVAL (line value) Input Command

Purpose: To compute the coordinates of a node based on its position along the line **L** connecting two points in space. The position of the node on line **L** is defined by specifying the value of one of the node's x-, y-, or z-coordinates as the independent variable and computing the remaining coordinate values to ensure that the node falls on line **L**.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined node are affected.

PCOM _____ input parameters _____

<for 2D models>
LVAL *i j ncomp value x1 y1 x2 y2 icomp*

<for 3D models>
LVAL *i j k ncomp value x1 y1 z1 x2 y2 z2* (*icomp*)

i, j, k = (I) Nodal ijk-indices of the node whose coordinates are set by this command.
No default.

ncomp = (C) Independent variable coordinate.
Any of: X, Y or Z
No default

value = (F) Value of the *ncomp* coordinate used to define all other coordinates based on its location on line **L** running connecting Point 1 to Point 2.
Default: 0.0

x1, y1, z1 = (F) X-, y- and z-coordinates defining Point 1.
Default: 0.0

x2, y2, z2 = (F) X-, y- and z-coordinates defining Point 2.
Default: 0.0

icomp = (C) Optional parameter. Used to define only a single coordinate component for the node. The other two coordinate components are left unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

PRNT (print) Input Command

Purpose: To print the values of variables in data arrays controlled by the Data Manager. Useful for printing the x-, y-, and z-coordinate values generated using Build commands.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: Prints Data Manager Array values at the time the command is input.

PCOM _____ input parameters _____

PRNT *dataname option ibegin iend jbegin jend kbegin kend*

dataname = (C) Name of data array containing elemental or nodal variables to be printed. No default.
For example: XCRD - x-coordinate array
YCRD - Y-coordinate array
ZCRD - Z-coordinate array

option = (C) Defines the print option for this command.
Any of: I = print J-K plane of values for each I-index
J = print I-K plane of values for each J-index
K = print I-J plane of values for each K-index
MNMX = print the min and max values for the specified region of the data group
Default = K

ibegin, iend = (I) Beginning and ending I-indices of nodes to be printed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for nodes to be printed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices for nodes to be printed. Input only for 3D models.
Default: *kbegin* = 1, *kend* = maximum K

Note: The **PRNT** command can be used to print coordinate values after any **Build** coordinate generation command in order to verify the command is working as expected.

Example: The following command prints the x-ccordinate values for the group of nodes with I-index in the range 1 to 10, j-index in the range 11 to 15, and k-index in the range of 28 to 30.

PRNT XCRD K 1 10 11 15 28 30

QUAD (quad region) Input Command

Purpose: The **QUAD** command computes the coordinates of all nodes forming a 2D plane in an ijk-region of the grid using bilinear interpolation based on the coordinates of the four corner nodes of the region. The corner node coordinates must already have been defined at the time the command is input.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected..

PCOM _____ input parameters _____

QUAD *(axisname) ibegin iend j*
begin jend (kbegin kend) icomp

axisname = (C) Optional input parameter. The name of a local axis system previously defined by the **AXIS** command group. This option is used only to perform the quad operation in a cylindrical or spherical coordinate system, causing the interpolation to work directly with the curvilinear coordinates of the local system. This produces filled grid lines that are curved instead of straight. If not input, the quad operation works in the global Cartesian system.

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates will be computed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates will be computed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional parameters input only for 3D models. The beginning and ending K-indices of the nodal region whose coordinates will be computed.
Default: *kbegin* = 1, *kend* = maximum K

icomp = (C) Optional parameter. Used to define only a single coordinate component for the node. The other two coordinate components are left unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

Note: The quad region specified by the *ibegin, iend,* etc indices should form a 2D surface in ijk-space. Therefore, if *ibegin* ≠ *iend*, and *jbegin* ≠ *jend,* then *kbegin* = *kend*, etc.

HEX (HEX region) Input Command

Purpose: To compute the coordinates of all nodes forming a 3D hexahedron in an ijk-region of the grid using trilinear interpolation based on the coordinates of the eight corner nodes of the region. The corner node coordinates must already have been defined at the time the command is input.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected..

PCOM input parameters

HEX *(axisname) ibegin iend jbegin jend kbegin kend icomp*

axisname = (C) Optional input parameter. The name of a local axis system previously defined by the **AXIS** command group. This option is used only to perform the hex operation in a cylindrical or spherical coordinate system, causing the interpolation to work directly with the curvilinear coordinates of the local system. This produces filled grid lines that are curved instead of straight. If not input, the hex operation works in the global Cartesian system.

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates will be computed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates will be computed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional parameters input only for 3D models. The beginning and ending K-indices of the nodal region whose coordinates will be computed.
Default: *kbegin* = 1, *kend* = maximum K

icomp = (C) Optional parameter. Used to define only a single coordinate component for the node. The other two coordinate components are left unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

RADI (Radius) Input Command

Purpose: To position a group of nodes at constant radius from either a cylindrical or spherical coordinate system. The original position of each node and the radius vector for the cylindrical or spherical coordiante system are used to define the ray along which the node is moved to reach the proper radius.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM input parameters

RADI *axisname radius ibegin iend jbegin jend kbegin kend*

axisname = (C) Name of a local axis system previously defined by the **AXIS** command group. The coordinate system must be cylindrical or spherical.

radius = (F) Radius at which all nodes are positioned.

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates are computed.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates are computed.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional parameters input only for 3D models. The beginning and ending K-indices of the nodal region whose coordinates are computed.
Default: *kbegin* = 1, *kend* = maximum K

Purpose: Tp read in the coordinates contained on the Grid Input File and assign these values to coordinates of nodes bounded by the input nodeal indices.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM input parameters

READ *form ibegin iend jbegin jend (kbegin kend)*

form = (C) File format for the Grdi Input File.
Any of: BIN or FRMT
BIN = binary file
FRMT = formatted file
Default = BIN

ibegin, iend = (I) Beginning and ending I-indices of the nodes to be read in.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for the nodes to be read in.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional parameters input only for 3D models. The beginning and ending K-indices for nodes to be read in.
Default: *kbegin* = 1, *kend* = maximum K

Note: Coordinates are read sequentially from the Grid Input File. Therfore, the first coordinates stored on the file are stored for node (ibegin, jbegin, kbegin), the second set of coordinates are stored for node (ibegin+1, jbegin, kbegin), etc.

RFIL (repetitive fill) Input Command

RFIL

Purpose: The **RFIL** command is similar to the **FILL** command except that it computes the coordinates of internal nodes in the i-, j-, or k-direction based on a series of beginning and ending nodes defined by the beginning, ending, and increment values of the *fillindex* indices specified. Therefore it does not simply fill between the bounding indices of the region indicated but does a series of fills between internal nodes within the region. The coordinates of all nodes used to define the interpolation bounds must already have been defined. *fillindex* defines the indices direction along which to interpolate. If *fillindex* is set to I, then: for each line of nodes specified by a (*j,k*) pair defined by the *jbegin, jend, jinc,* and *kbegin, kend, kinc* values input, the nodes between (*ibegin,i,k*) and (*ibegin+iinc,j,k*) are computed based on the coordinates of these two bounding nodes, the coordinates of nodes between (*ibegin+iinc,j,k*) and (*ibegin*+2***iinc,j,k*) are computed based on the coordinates of these two bounding nodes, etc. When *fillindex* is set to J or K the analogous procedure is used to perform the repetitive fill operation in those indices directions. This command is useful for computing the coordinates of interior groups of nodes on an interface on which key interior coordinates have been assigned with the **COPY** command.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM _____ input parameters _____

RFIL *fillindex ibegin iend iinc jbegin jend jinc (kbegin kend kinc) icomp*

fillindex = (C) Grid index to fill along. Any of: I, J, or K.
For example, if *fillindex*=I, the coordinates of all nodes within the ijk window having *ibegin*< I-index < *iend* are computed using interpolation based on the coordinates of the bounding nodes having
I-index=*ibegin* and I-index=*iend*. No default.

ibegin, iend, iinc = (I) Beginning and ending I-indices of the bounding nodes for the region of nodes to be filled.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jen, jinc = (I) Beginning and ending J-indices of the bounding nodes for the region of nodes to be filled.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend, kinc = (I) Optional input parameters input only for 3D models. The beginning and ending K-indices of the bounding nodes for the region of nodes to be filled.
Default: *kbegin* = 1, *kend* = maximum K

icomp = (C) Optional parameter. Used to define only a single coordinate component for the node. The other two coordinate components are left unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

ROT (rotate) Input Command

ROT

Purpose: To rotate the coordinates of the group of nodes by a specified angle about a specified axis.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected..

PCOM _____ input parameters _____

ROT *[axisname] delangle ibegin iend jbegin jend (kbegin kend)*

axisname = (C) Name of the axis to rotate about. May be any of X, Y, Z, or any of the local axes defined by the **AXIS** command, i.e., if a local coordinate system named SYS1 has previously been defined, then *axisname* could be set equal to any of the following constructs: SYS1, SYS1 X, SYS1 Y, or SYS1 Z. SYS1 without an axis name following defaults to SYS1 X..
No default.

delangle = (F) Rotation angle in degrees.
Default: 0.0

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates will be rotated.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates will be rotated.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional input parameters input only for 3D models. The beginning and ending K-indices of the nodal region whose coordinates will be rotated.
Default: *kbegin* = 1, *kend* = maximum K

RPOS (relative position) Input Command

Purpose: To assign the coordinates of Node1 according to the rule that the coordinates of Node1 have the same relative position to Node2 as Node3 has to Node4.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined node are affected.

PCOM _____ input parameters _____

<for 2D models>
RPOS *i1 j1 i2 j2 i3 j3 i4 j4 icomp*

<for 3D models>
RPOS *i1 j1 k1 i2 j2 k2 i3 j3 k3 i4 j4 k4 icomp*

i1, j1, k1 = (I) The ijk-indices of Node1

i2, j2, k2 = (I) The ijk-indices of Node2

i3, j3, k3 = (I) The ijk-indices of Node3

i4, j4, k4 = (I) The ijk-indices of Node4

icomp = (C) Optional parameter. Used to define only a single coordinate component for the node. The other two coordinate components are left unchanged.
Any of: X, Y or Z.
Default: all coordinate components are set with this command.

SCAL (scale) Input Command

SCAL

Purpose: To scale all coordinates in an ijk-region of the model.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM input parameters

<for 2D models>
SCAL *xscale yscale ibegin iend jbegin jend*

<for 3D models>
SCAL *xscale yscale zscale ibegin iend jbegin jend kbegin kend*

xscale
yscale, zscale = (F) Scale factor to use for scaling the x-, y-, and z-coordinates, respectively.
Default = 1.0

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates are scaled.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates are scaled.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices of the nodal region whose coordinates are scaled.
Default: *kbegin* = 1, *kend* = maximum K

Purpose: To set the values of variables in data arrays controlled by the Data Manager. Useful for setting the x-, y-, or z-coordinate values of a large group of nodes to a single value. Standard **SET** command as used in **PZFlex**.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM input parameters

<for 2D models>

SET *dataname value ibegin iend jbegin jend*

<for 3D models>

SET *dataname value ibegin iend jbegin jend kbegin kend*

dataname = (C) Name of data array containing the nodal coordinates set by this command. No default.
For example: XCRD - x-coordinate array
YCRD - Y-coordinate array
ZCRD - Z-coordinate array

value = (F) Value to set the coordinates to.
Default = 0.0

ibegin, iend = (I) Beginning and ending I-indices of the nodes to be set.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for the nodes to be set.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices for nodes to be set.
Default: *kbegin* = 1, *kend* = maximum K

Note: The **SET** command is very useful for setting the coordinate values for a part of the model that has been copied using the COPY command.

Example: The following command sets the z-coordinates to 25.8 for the group of nodes with I-index in the range 1 to 10, J-index in the range 1 to 15, and K-index in the range of 30 to 30.

SET ZCRD 25.8 1 10 1 15 30 30

SHFT (shift) Input Command

SHFT

Purpose: To shift all nodal coordinates in an ijk-region of the grid by a specified amount.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM input parameters

<for 2D models>
SHFT *xshift yshift ibegin iend jbegin jend*

<for 3D models>
SHFT *xshift yshift zshift ibegin iend jbegin jend kbegin kend*

xshift
yshift, zshift = (F) Amount to shift the nodal coordinates by in the x-, y- and z-directions, respectively.
Default = 1.0

ibegin, iend = (I) Beginning and ending I-indices of the nodes to be shifted.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for the nodes to be shifted.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Beginning and ending K-indices for nodes to be shifted.
Default: *kbegin* = 1, *kend* = maximum K

SOR (solid/surface of revolution) Input Command

Purpose: Using a set of starting nodes, to create a circular line from a starting point, a surface of revolution from a starting line, or a solid of revolution from a starting plane of nodes. The coordinates of the starting nodes are not affected by this command. The coordinate values of the starting nodes are revolved around a specifed axis to create a rotated set of coordinates with a constant arc spacing specified by the user.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected..

PCOM input parameters

SOR *[axisname] circindex delangle ibegin iend jbegin jend (kbegin kend)*

axisname = (C) Name of the axis to revolve the geometry about. May be X, Y, Z, or any of the local axes defined by the **AXIS** command, i.e., if a local coordinate system named SYS1 has previously been defined, *axisname* could be set equal to any of the following constructs: SYS1, SYS1 X, SYS1 Y, or SYS1 Z. SYS1 without an axis name following defaults to SYS1 X..
No default.

circindex = (C) Nodal index that varies in the circumferential direction. Any of: I, J, or K.
No default.

delangle = (F) Incremental arc angle between successive nodes [degrees].
Default: 0.0

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates will be generated.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates will be generated.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional input parameters. Input only for 3D models. The beginning and ending K-indices of the nodal region whose coordinates will be generated.
Default: *kbegin* = 1, *kend* = maximum K

STOP Input Command

STOP

Purpose: To terminate the processing of a job in an orderly manner.

Use Is: Optional.

Multiple Use: Not allowed.

Order Dependence: No commands following a **STOP** command are processed.

PCOM	input parameters
STOP	-none-

Notes: 1. If a **STOP** command is encountered in the middle of the input stream, no input commands following the **STOP** command are processed.

2. If no **STOP** command is found and the end-of-file for the Job Input File is encountered, a **STOP** command is automatically generated, resulting in a normal termination of job processing.

Purpose: To adjust the position of nodes in an ijk-region of the grid to lie on surface **S** by adjusting a single, user-specified, global coordinate of the nodes. Surface **S** is defined as a constant coordinate contour in space.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined node are affected.

PCOM input parameters

SVAL *[axisname] value ncomp ibegin iend jbegin jend (kbegin kend)*

axisname = (C) *axisname* and *value* define the surface, **S**. *axisname* defines the name of the local coordinate component that defines the surface contour. For Cartesian systems, set to X, Y, or Z for global axes or reference previously defined local axes (using the**AXIS** command) using the following constructs. For local system SYS1: SYS1 X, SYS1 Y, or SYS1 Z. For cylindrical or spherical coordinate systems, the radial coordinate is always assumed.
No default.

value = (F) Value of the the *axisname* coordinate component that defines the surface contour of interest
Default: 0.0

ncomp = (C) Dependent variable coordinate. This coordinate is adjusted to place each node on the specified surface. The remaining coordinates are not changed. Any of: X, Y, or Z
No default

ibegin, iend = (I) Beginning and ending I-indices of the nodal region whose coordinates are placed on the surface.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices of the nodal region whose coordinates are placed on the surface.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional input parameters, input only for 3D models. The beginning and ending K-indices of the nodal region whose coordinates are placed on the surface.
Default: *kbegin* = 1, *kend* = maximum K

Examples:

Move nodes having I-index in the range of 1 to 10, J-index in the range of 1 to 5, and K-index in the range of 11 to 11 to the surface defined by y-coordinate = 20. Only the y-coordinate of the nodes is changed.

SVAL Y 20. Y 1 10 1 5 11 11

Move nodes having I-index in the range of 1 to 10, J-index in the range of 1 to 5, and K-index in the range of 11 to 11 to the cylindrical surface defined by radius=11.5 for the user defined local cylindrical system named CYL1. Only the x-coordinate of the nodes is changed to place each node on the surface.

SVAL CYL1 11.5 X 1 10 1 5 11 11

SWTH (switch) Input Command

Purpose: To switch the values of two coordinate components of a group of nodes.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: All subsequent commands that access the coordinates of the defined nodes are affected.

PCOM input parameters

SWTH *iaxes ibegin iend jbegin jend (kbegin kend)*

iaxis = (C) Character string defining the two coordinate components to be switched, i.e., if set to YZ, the y-coordinate and z-coordinate values of the nodes are switched. Any of: XY, YZ, or XZ
No default.

ibegin, iend = (I) Beginning and ending I-indices of the nodes to be switched.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for the nodes to be switched.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional input parameters input for 3D models. The beginning and ending K-indices for nodes to be switched.
Default: *kbegin* = 1, *kend* = maximum K

TERM (terminal) Input Command

Purpose: To control the routing of standard input and output files to the terminal or to the Job Input and Job Output Files.

Use Is: Optional.

Multiple Use: Allowed.

Order Dependence: None.

PCOM _____ input parameters _____

TERM *inoption outoption*

or

T *inoption outoption*

inoption = (C) Option controlling standard input. Either ON or OFF.
If ON, the user must provide input commands from the terminal keyboard in interactive mode.
If OFF, input commands are read from the Job Input File.
Default: toggle from current state. If currently ON, default is OFF. If currently OFF, default is ON.

outoption = (C) Option controlling standard output . Either ON or OFF.
If ON, standard output from the code is written to the user's terminal screen.
If OFF, standard output is written to the Job Output File.
Default = *inoption*.

Notes:
1. Standard output displayed on a user's terminal is not recorded in the Job Output File.
2. If the user switches the standard input from the Job Input File to the terminal keyboard and later switches input back to the Job Input File, processing of commands from the Job Input File continues where it had left off.

Examples:

TERM /* toggle ON to OFF or OFF to ON

TERM OFF ON /* input is read from Job Input File, output is displayed at terminal

TERM ON OFF /* input from keyboard, output to Job Output File

TITL (title) Input Command

Purpose: Standard **TITL** command as used in **PZFlex**. Currently, this command serves no function except to act as a descriptive label for the command file.

Use Is: Optional but recommended.

Multiple Use: Not allowed.

Order Dependence: None. Recommended to be first command.

PCOM input parameters

TITL <--*id*--> <----------*title*---------->

id = (C) Up to 8-character job identifier used to tag output results associated with the job. If internal blanks are desired, must be enclosed in single quotes. No default.

title = (C) 60-character job title. This input character string need not be enclosed in single quotes even if it includes blanks. All characters following the *id* field are assumed to be part of *title*.
No default.

WRIT (write) Input Command

WRIT

Purpose: To write the current coordinate information for the entire grid or some ijk-region to the Grid Output File.

Use Is: Required to produce coordinate data file from BUILD.

Multiple Use: Not allowed.

Order Dependence: Should follow all coordinate generation commands.

PCOM _____ input parameters _____

WRIT *form ibegin iend jbegin jend (kbegin kend)*

form = (C) File format for the Grid Output File.
Any of: BIN or FRMT
BIN = binary file
FRMT = formatted file
Default = BIN

ibegin, iend = (I) Beginning and ending I-indices of the nodes to be written out.
Default: *ibegin* = 1, *iend* = maximum I

jbegin, jend = (I) Beginning and ending J-indices for the nodes to be written out.
Default: *jbegin* = 1, *jend* = maximum J

kbegin, kend = (I) Optional input parameters input for 3D models. The beginning and ending K-indices for nodes to be written out.
Default: *kbegin* = 1, *kend* = maximum K

Note: Coordinates are written sequentially to the Grid Output File (the file name containing the text string: bldgrdo). Therefore, the first coordinates written to the file are for node (*ibegin, jbegin, kbegin*), the second set of coordinates written are for node (*ibegin+1, jbegin, kbegin*), etc.

Example : The following command contained in file bldinp.model writes all coordinates to the Grid Output File (bldgrdo.model) in binary format:

WRIT

The following command, contained in file bldinp.run5, writes the coordinates for nodes with I-indices ranging from 11 to 20, j-indices ranging from 1 to 50, and
k-indices ranging from 1 to 5 to the Grid Output File (bldgrdo.run5) in formatted text format:

WRIT FRMT 11 20 1 50 1 5\