



ReSHAPE Reference Manual

(Revision: 28 August, 2008, Edited by Ryan Adams)

Version A08

ADVEA Engineering Pty. Ltd.
7 Calderwood Avenue,
Wheelers Hill 3150,
Victoria, Australia.

Tel: +61 3 9558 6666
Fax: +61 3 9558 6677

E-mail: info@advea.com
<http://www.advea.com>

The information in this document is subject to change without notice and should not be interpreted as a commitment by ADVEA Engineering Pty. Ltd.

ADVEA DOES NOT ASSUME ANY RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT OR ANY CONSEQUENCES THESE ERRORS MAY CAUSE AND/OR HAVE.

The software described in this document is furnished under license and may be used or copied only in accordance with the terms of such license.

No part of this document may be reproduced in any shape or form without prior agreement in writing from ADVEA Engineering Pty. Ltd.

No part of this document may be distributed in any way without prior agreement in writing from ADVEA Engineering Pty. Ltd.

© ADVEA Engineering Pty. Ltd., 2007

Printed in Australia

All Rights Reserved

ReSHAPE is a registered trademark of ADVEA Engineering Pty. Ltd.

MSC.NASTRAN is a registered trademark of MSC Software.

MEDINA is a registered trademark of T-Systems ITS GmbH.

HyperMesh is a registered trademark of Altair Engineering.

This manual is only part of the documentation available from ADVEA Engineering Pty. Ltd. for ReSHAPE. The complete set of ReSHAPE manuals consists of:

ReSHAPE Reference Manual

A dictionary of the input and output commands, and run time commands for general operation of ReSHAPE. It is intended as the basic reference document.

Starting with ReSHAPE

Intended for the novice of ReSHAPE to provide a quick introduction into using ReSHAPE for the solution of practical problems. Input files (in the NASTRAN *.bdf format) and command files are provided for most of the problems.

ReSHAPE Theory Manual

Explains the basics of mathematical methods used in ReSHAPE.

Case Studies

Can be downloaded from the web site, and contain challenging, large and practical problems that have been solved using ReSHAPE.

ReSHAPE/Example Data Base

An extensive data base of examples is provided, containing input files (*.bdf) and command files (*.cmf) for most of the problems solved in *Starting with ReSHAPE*.

Files can be obtained from the 'Examples' tab in the user interface.

The following code has been used for the file names:

exccppxa.<ext>, where

cc – chapter number with a leading 0 if less than 10

pp – paragraph number with a leading 0 if less than 10

xx – problem number with a leading 0 if less than 10

a – optional letter.

i.e., the example 5.2.1 will have the input file name **ex050201.bdf** and the command file name **ex050201.cmf** and maybe **ex050201a.cmf** or others.

CONTENTS

RESHAPE OVERVIEW.....	VI
HEADER.....	1
<i>RESHAPE</i>	2
SET BLOCK.....	3
<i>ESET</i>	4
<i>GSET</i>	5
<i>NSET</i>	6
CONTROL BLOCK	7
<i>CONTROL</i>	8
<i>2DMODEL</i>	9
<i>ANALYSIS</i>	10
<i>CPARAMETER</i>	12
<i>NASTRAN</i>	13
<i>OOC</i>	14
<i>REPLAY</i>	15
<i>SOLVER</i>	16
<i>STEPS</i>	17
<i>SUBCASE</i>	18
<i>SUPER</i>	19
RESPONSES BLOCK	20
<i>RESPONSES</i>	21
<i>COMPLIANCE</i>	23
<i>DISPLACEMENT</i>	24
<i>EXTERNAL</i>	29
<i>MASS, VOLUME, AREA, LENGTH</i>	30
<i>MODE, BUCKLE</i>	31
<i>REACTION</i>	32
<i>STRESS</i>	34
SIDE CONDITIONS BLOCK.....	36
<i>SIDECONDITIONS</i>	37
<i>ALIMITS</i>	38
<i>AVERAGE</i>	39
<i>MASTERSLAVE</i>	41
<i>NLIMITS</i>	42
<i>RLIMITS</i>	43
<i>TAVERAGE</i>	44
PROCESS BLOCK	46
PROCESS(RAW) BLOCK.....	47
<i>DOMAIN</i>	48
<i>LOCKED</i>	49
<i>NORMAL</i>	50
<i>PASSIVE</i>	51
<i>UNLOCKED</i>	53
PROCESS(FUNCTION) BLOCK	54
<i>DOMAIN</i>	55
<i>FCONTROL</i>	57
<i>FPARAMETER</i>	59

<i>FPLANE</i>	61
<i>LOCKED</i>	62
<i>NORMAL</i>	63
<i>PASSIVE</i>	64
<i>UNLOCKED</i>	65
PROCESS(GEOMETRY) BLOCK.....	66
<i>DOMAIN</i>	67
<i>GCORNERS</i>	68
<i>GLINE</i>	69
<i>GPARAMETER</i>	70
<i>GRIB</i>	71
<i>LOCKED</i>	72
<i>NORMAL</i>	73
<i>PASSIVE</i>	74
<i>UNLOCKED</i>	76
PROCESS(VECTOR) BLOCK.....	77
<i>DOMAIN</i>	78
<i>LOCKED</i>	79
<i>NORMAL</i>	80
<i>PASSIVE</i>	81
<i>UNLOCKED</i>	82
<i>VCREATE</i>	83
<i>VINPUT</i>	84
PROCESS(EVOLUTION) BLOCK.....	86
<i>DOMAIN</i>	87
<i>LOCKED</i>	88
<i>NORMAL</i>	89
<i>PASSIVE</i>	90
<i>UNLOCKED</i>	92
<i>EPARAMETER</i>	94
PROCESS(SIZING) BLOCK.....	95
<i>SDOMAIN</i>	96
<i>SPARAMETER</i>	97
PROCESS(TOPOLOGY) BLOCK.....	98
<i>TDOMAIN</i>	99
<i>TPARAMETER</i>	100
APPENDICES	102
<i>ReSHAPE Structure Overview</i>	103
<i>FORMAT OF ReSHAPE COMMANDS</i>	109
<i>Running ReSHAPE</i>	110
<i>Results file codes</i>	114
<i>ReSHAPE ⇔ NASTRAN commands</i>	115
<i>NASTRAN elements and ReSHAPE commands</i>	120

ReSHAPE Overview

The program ReSHAPE can improve the design of the most complicated structures by modifying their shape. ReSHAPE can perform many design improvements by changing the component shape in order to:

- reduce weight
- increase stiffness
- reduce stresses
- increase contact forces
- reduce or increase frequencies of vibration
- increase buckling resistance

These objectives can be achieved under a variety of constraints.

The design improvement process is iterative, based on the linear finite-element method. As such, only linear or quasi-linear problems can be solved by ReSHAPE. The iterative improvement is based on the calculation of sensitivities in the finite-element nodes.

The nodal sensitivities may also be used before the start of the iterative process for the identification of sensitive regions of the structure. This helps avoiding futile attempts in trying to improve a given model in insensitive areas, or attempting to modify models of inherently bad design.

ReSHAPE uses NASTRAN input files (*.bdf, *.dat or *.nas) and creates NASTRAN, MEDINA or HYPERMESH output files (*.op2, *.bof or *.res). The output files allow the results to be visually displayed in the post-processing phase.

ReSHAPE can be used in two non-iterative and one iterative solution procedures:

1. ANALYSIS provides all the basic elements and features of a linear finite-element program. (See Appendix 4 for acceptable NASTRAN bulk commands).
2. SENSITIVITY calculates the sensitivity of the model to a design objective. Such sensitivities are called “raw” nodal sensitivities, i.e. no mesh-geometry relationship (like parametric co-ordinates of splines, basis vectors or influence functions) is used for “smoothing”. There is no iteration involved in this process. Consequently, the only constrained responses that can influence the results are those required to keep original values (parameter `bound=hold` in constraint commands).
3. IMPROVEMENT has seven different iterative design improvement processes, which modify the shape of the model by a gradient method, using the sensitivities calculated at each step of the iterative process. Apart from `PROCESS(RAW)`, the mesh “smoothness” is preserved. Additionally, two element based processes are available: `PROCESS(SIZING)` and `PROCESS(TOPOLOGY)`.

ReSHAPE can employ seven different improvement process:

1. RAW moves each node independently according to the calculated raw sensitivities.
2. GEOMETRY recalculates cubic and bi-cubic geometric entities of the original model by the best fit to the nodal changes based on raw sensitivities
3. FUNCTION uses influence functions for smooth shape changes.
4. VECTOR uses pre-calculated basis vectors for smooth shape changes.
5. EVOLUTION automatically generates possible shape changes based on the sensitivity vector.
6. SIZING changes thickness of shell element
7. TOPOLOGY redistributes material

All of these processes can be solved as a standard analysis, or using the NASTRAN inertia relief method PARAM, INREL, -2. In the case of VECTOR and EVOLUTION, inertia relief is automatically activated when there are no locked nodes.

HEADER

HEADER

Commands: *none*

The RESHAPE header has two purposes:

- To start the command file and indicate the type of processing to be executed.
- To serve as a header of RESHAPE SETS.

RESHAPE

Header for ReSHAPE Command File and SET Block

RESHAPE { { ANALYSIS
SENSITIVITY
IMPROVEMENT } }

The RESHAPE command is always the first command in the command file. It initiates reading and processing of individual blocks.

Parameters:	Meaning
ANALYSIS	Specifies to the analysis procedure should be used (not required; see Remarks 2,3,4).
SENSITIVITY	Specifies that the calculation of weighted raw sensitivities in the domain specified in the PROCESS block should be performed.
IMPROVEMENT	Specifies that an iterative calculation using one of the following methods specified (as a parameter of the PROCESS command): RAW, FUNCTIONS, GEOMETRY, VECTORS, EVOLUTION, SIZING or TOPOLOGY.

Examples:

```
reshape(sensitivity)
reshape(impr)
```

Remarks:

1. If no ReSHAPE command file is provided, ReSHAPE will execute an FE analysis utilising all required information (analysis type, load sub-cases, etc.) from the NASTRAN input file.
2. If a command file (*.cmf) is used then the ReSHAPE command parameter ANALYSIS must be input if analysis is required. In this case all subsequent commands will be ignored with the exception of the 2DMODEL, ANALYSIS and SUBCASE commands in the CONTROL block.
3. For the generation of basis vectors, ReSHAPE must be run using RESHAPE (ANALYSIS) in the command file. The command VCREATE must be present in the PROCESS(VECTOR) block, and an ANALYSIS directive command must be in the CONTROL block (eg, ANALYSIS(STATIC)).
4. GSET, NSET and ESET commands must follow immediately after the RESHAPE command.

SETS

SET block

Commands: *ESET*
 GSET
 NSET

The commands in the RESHAPE SET block provide coupling between NASTRAN set commands and RESHAPE set commands. This is necessary, because NASTRAN does not provide information about a set, whether it is a node set or an element set.

For PROCESS(GEOMETRY) (see PROCESS(GEOMETRY) block), the user must provide information about splines and surfaces to be created. This is achieved by assigning NASTRAN element sets to ReSHAPE GSETS. To each GSET, RESHAPE fits a cubic spline for beams and bi-cubic patches (surfaces) for shells, and calculates the parametric co-ordinates. Only three and four sided regions of elements can be used for surfaces, and adjacent GSET surfaces must share coincident corner nodes.

SETS

ESET

Element Set Definition

ESET(setname)=sids

This command imports a NASTRAN SET definition into a labeled ReSHAPE element set.

Parameters	Meaning
setname	Identifier name (character) of the element set (maximum characters = 16). (Remark 1).
sids	Set identification numbers (integer > 0). Sets must be defined by single integers (i.e. $i_1, i_2, i_3, \dots, i_n$ where $i_1 < i_2 < i_3 < \dots < i_n$) (Remark 2).

Examples:

```
eset(crankARM)=7,8  
eset (LEFT_HALF)=2  
eset(BUSHING)=99 thru 120
```

Remarks:

1. No spaces are allowed in the identifier name of an ESET. The identifier name must be unique within the input file. Identifier names are case sensitive. Supported characters are a to z, A to Z, 1 to 9, and + - _ @ # % &.
2. The sids refers to the identification numbers of the SETs defined in the case control section in NASTRAN. The identification numbers should refer to elements in the BULK DATA section. The keyword EXCEPT is not supported, therefore appropriate changes must be made to NASTRAN definitions of a SET in cases where EXCEPT has been used. Most pre-processors do not create SETs with the EXCEPT keyword.
3. As many ESET sets as needed can be defined.
4. As many NASTRAN sets as necessary may be combined into an ESET.

SETS

GSET

Geometry Element Set Definition

GSET(setname)=sids

GSET allocates NASTRAN element sets into labeled ReSHAPE element sets to be used for the generation of cubic splines for PROCESS(GEOMETRY).

Parameters	Meaning
setname	Identifier name (character) of the element set (maximum characters = 16). (Remark 1 for ESET command on previous page).
sids	Set identification numbers (integer > 0) Sets must be defined by single integers (i.e. $i_1, i_2, i_3, \dots, i_n$ where $i_1 < i_2 < i_3 < \dots < i_n$). (Remark 1 below).

Examples:

gset(patch11)=12,14
gs(FRONT)=1,5,7 thru 29

Remarks:

1. The sids refer to the identification numbers of the SETs defined in the case control section in NASTRAN, which refer to elements in the BULK DATA section. The keyword EXCEPT is not supported, therefore appropriate changes must be made to NASTRAN definitions of a SET in cases where EXCEPT has been used.
2. Each NASTRAN element set has to define a region of elements suitable to be fitted with cubic splines. The splines should have one edge for 1D elements and three or four clearly defined edges for 2D elements (ie. triangle or quadrilateral). 3D blocks and regions consisting of 3D elements are not supported.
3. The corner nodes of each spline region must coincide with the corner nodes of any and all other adjacent spline regions
4. In cases where ReSHAPE cannot automatically interpret the user-defined geometry, the nodes which are to be regarded as the corners of the region may be specified separately using GCORNER command in the PROCESS(GEOMETRY) block. An error is signaled if the re-meshed approximation to the element region results in the movement of nodes more than a set tolerance from their original position.
5. The user is strongly advised to display the output file *.gsh in a post-processor (written in nodal displacement format) to see corrections to the mesh due to spline fitting.
6. Elements defined by the GSET command can be used as ESETs in other processes.

SETS

NSET

Node Set Definition

NSET(setname)=sids

This command group predefines NASTRAN SET definitions into labeled ReSHAPE node sets

Parameters	Meaning
setname	Identifier name (character) of the node set (maximum characters = 16) (Remark 1)
sids	Set identification numbers (integer > 0). Sets must be defined by single integers (i.e. $i_1, i_2, i_3, \dots, i_n$, where $i_1 < i_2 < i_3 < i_n$) (Remark 2).

Examples:

```
nset(PLATE_HOLES)=1,2,3  
nset (SURFACENODES)=157  
nset(SURFACENODES)=1,4,5,20,88
```

Remarks:

1. No spaces are allowed in the identifier name of an NSET. The identifier name must be unique within the input file. ReSHAPE treats identifier names as being case sensitive. Supported characters are a to z, A to Z, 1 to 9, and + - _ @ # % &
2. The sids refer to the identification numbers of the SETs defined in the case control section in NASTRAN. The identification numbers should refer to nodes or elements in the BULK DATA section. The keyword EXCEPT is not supported, therefore appropriate changes must be made to NASTRAN definitions of a SET in cases where EXCEPT has been used. Most pre-processors do not create SETs using the EXCEPT keyword.
3. If a ReSHAPE command refers to an element set, when a node set is the required input, the nodes of the elements are extracted.
4. As many NSET sets as needed can be defined.
5. As many NASTRAN sets as necessary may be combined into an NSET.

CONTROL

CONTROL block

Commands:

2DMODEL
ANALYSIS
CPARAMETER
NASTRAN
OOO
REPLAY
SOLVER
STEPS
SUBCASE
SUPER

The commands in the CONTROL block provide the user with various features for controlling the design process. They are described in detail on the following pages.

2DMODEL is used when a model is fully defined in the xy-plane only.

ANALYSIS is applicable only to PROCESS(VECTOR).

PROCESS(SIZING) uses 2DMODEL, STEPS, SUBCASE and CPARAMETER commands only.

CONTROL

CONTROL

Header of the Control Block

CONTROL

Examples:

control

Remarks:

1. The CONTROL command is compulsory for the use of an improvement process i.e., RAW, GEOMETRY, VECTORS, or FUNCTION.

CONTROL

2DMODEL

Model is Two-dimensional (in XY-plane)

2DMODEL

Examples:

2Dmodel

2dmodel

Remarks:

1. The model must be fully defined in the xy-plane (including loads).
2. Degrees of freedom TZ, RX, RY and RZ are fixed for both the ANALYSIS and IMPROVEMENT processes.
3. Sensitivities in the internal nodes are not calculated when in 2DMODEL mode, i.e. command PASSIVE (INTERNAL) is activated.
4. Internal nodes are re-meshed (in FUNCTION, VECTOR and GEOMETRY processes).
5. Two-dimensional static or vibration analysis can only be achieved by referencing a *.cmf file containing the 2DMODEL command.

ANALYSIS

Select the Type of Analysis

$$\mathbf{ANALYSIS} \left\{ \left(\begin{array}{l} \mathit{STATIC}[, \mathit{STRESS} = \mathit{stress}] \\ \mathit{VIBRATION}[, \mathit{FMIN} = \mathit{f\ min}][, \mathit{NMODES} = \mathit{n}] \\ \mathit{BUCKLING}[, \mathit{NMODES} = \mathit{n}] \end{array} \right) \right\}$$

Optional selection of the analysis type for the RESHAPE(ANALYSIS) header.

Parameters	Meaning
STATIC	Linear static analysis will be solved (SOL=101 in NASTRAN)
STRESS=stress	Stress result to put into results files (eg, vmmax). See keyword 'Stress' in chapter on Objectives for a full list of available stress results. Default: STRESS=vmmax.
VIBRATION	Vibration modal analysis will be solved (SOL=103)
BUCKLING	Buckling modal analysis will be solved (SOL=105)
FMIN=frequency	Minimum frequency to be calculated. Default is 0.0 Hz. This may be omitted unless the model is not fully fixed. If the model is not fully constrained, a value greater than 0.0 Hz must be specified (used as a frequency shift). Frequency is in Hz (real).
NMODES=n	Number of modes to be calculated. Default is eight (8) for VIBRATION and one (1) for BUCKLING.

Examples:

```
analysis(vib,nm=4)
analysis(buckling)
analysis(static,stress=sxxg)
```

Remarks:

1. If ANALYSIS command is not present, the default is the method used in the NASTRAN Executive Control Statement SOL. ReSHAPE supports SOL=101, 103, and 105. The user has to provide the NASTRAN command EIGRL in the NASTRAN input file if VIBRATION or BUCKLING is to be solved. For a STATIC analysis, the loads are selected either by one and only one load case present in the NASTRAN input file, or by the ReSHAPE command SUBCASE in the CONTROL block.

CONTROL

2. If the ANALYSIS command is present, ReSHAPE will ignore the SOL statement from the NASTRAN input file. The type of analysis will be dictated by the content of the ANALYSIS command.
3. If the keyword 'STRESS' is not supplied in an ANALYSIS(STATIC) command, then the default is STRESS=vmmax.
4. Unless the output file is specified in the Medina format, then post-processing of beam bending moments is not available. Bending moments can still be optimized, and the results can be read from the PLT9 file using the *-r plt9* flag in UNIX, or the menu item 'Create -> Results from *.op2 file...' in the PC version.
5. The analysis types STATIC and BUCKLING require loads. These can be selected from NASTRAN load cases by using the ReSHAPE command SUBCASE in the CONTROL block.
6. If no SUBCASE command is present in the ReSHAPE CONTROL block, the assumption is that there is only one SUBCASE in the NASTRAN input file. The loads and boundary conditions are then taken according the NASTRAN case controls SPC and LOAD. More than one SUBCASE in the NASTRAN input file will result in an error.
7. For analysis of VIBRATION or BUCKLING, the FMIN and NMODES parameters take precedence over the EIGRL NASTRAN command.

CONTROL

CPARAMETER

Control parameters

CPARAMETER(parameter) = value

Sets prescribed parameters for the overall control of the iterative improvement procedure. (Otherwise defaults are used).

Table 1 CPARAMETER Parameters and their default values

Parameter	Description	Default value
<i>MAXREDO</i>	Maximum number of iterations when attempting to achieve tolerance on an already achieved objective target, or the number of subdivisions after an unsuccessful iteration step.	4
<i>CONJUGATE</i>	Whether conjugate gradient will be applied after the first unsuccessful iteration.	YES
<i>INTERMEDIATE</i>	Output results for every iteration into the *.rp1 file for post-processing. Can be disabled for large models.	YES
<i>OBJCHANGEMIN</i>	Minimum change in the objective that is allowed to occur from one iteration to the next, given as an absolute value or percentage. Note: ReSHAPE will always try and do three iterations before this parameter takes effect.	0.1%

Examples:

cpar(maxredo)=6
cparameter(OBJCHANGEMIN)=5

CONTROL

NASTRAN

NASTRAN Output Control

NASTRAN(parameter) = value

Table 2 NASTRAN Parameters and their default values

Parameter	Description	Default value
<i>WARNINGS</i>	Sets the number of warnings to be printed in the log file for each unsupported NASTRAN command.	1
<i>BYPASS</i>	Ignores references to non-existent elements in NASTRAN sets.	NO (Do not bypass)

Examples:

nast(B)=no

nas(WARNINGS)=1

OOO = value**Table 3 SOLVER parameters**

Parameter	Description
<i>value</i>	Yes/No (Default: No) Use out-of-core routines to save factor to the disk if memory requirements are very high. Uses TAUCS to solve, and custom routines for the out-of-core data management.

Examples:

ooo=yes

CONTROL

REPLAY

Control of RePLAY Output File

$$\text{REPLAY} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$$

Parameters	Meaning
YES	RePLAY post-processing files are created (default)
NO	RePLAY post-processing files are not created.

Examples:

REPLAY=no
Replay=YES

Remarks:

1. For Replay=no, the files *.rp1 and *.rp2 are not created.
2. The use of replay=no can be used effectively when the user is working with large models and RePLAY post-processing results are not required. The replay=no command ables the user to reduce analysis time and memory usage.

SOLVER

Control of Simultaneous Equation Solver

SOLVER = type

Table 4 SOLVER parameters

Type	Description
<i>LDLT</i>	Use LDLT factorisation to solve the simultaneously equations. Essential for buckling, free vibration problems and inertia relief. Slow for constrained structural problems, use Cholesky instead. TAUCS LDLT solver.
<i>CHOLESKY</i>	Use LLT factorisation to solve the simultaneous equations. Much more sensitive to ill-conditioned matrix. Not recommended to be used to inertia relief, free vibration or buckling problems. TAUCS LLT solver.
<i>OOC</i>	Use out-of-core routines to save factor to the disk if memory requirements are very high. Uses TAUCS LDLT to solve, and custom routines for the out-of-core data management.

Examples:

solver=ldlt
solver=ooc

Remarks:

1. All solvers are available on all platforms.

CONTROL

STEPS

Control of Iterative Procedure

STEPS (parameter) = value

Table 5 STEP Parameters and their default values

Parameter	Description	Shape/Sizing defaults	Topology defaults
<i>NUMBER</i>	Number of steps	1	1
<i>SIZE</i>	Starting step size (%)	10% of CDA	50% of CDA
<i>MINIMUMSTEPSIZE</i>	Iteration of a process will stop below this value (%)	1% of CDA	1% of CDA
<i>MAXIMUMSTEPSIZE</i>	Step size will remain at this value if reached (%)	90% of CDA	N/A
<i>MULTIPLIER</i>	Increase step size after each iteration by percentage	1.05	1.05

Examples:

steps(n)=20
step(s)=3

Remarks:

1. CDA is the average characteristic dimension, and is different for each process. For TOPOLOGY it is the 'pseudo-density' of 1.0, for GEOMETRY it is the average side length of a patch, for other shape change processes it is the average element size.
2. STEPS command is ignored for both 'analysis' and 'sensitivity' procedures.
3. As many STEPS commands as necessary can be used.
4. Apart from parameter NUMBER, most problems should run efficiently with default values of other parameters.

CONTROL

SUBCASE

Selection of Analysis Subcase from the Input File

SUBCASE = n

Problem will be solved (in ANALYSIS, SENSITIVITY, or IMPROVEMENT mode) for the specified subcase, referring to a NASTRAN SUBCASE command.

Parameters:

Meaning

n Integer indicating the subcases to be analysed from the NASTRAN input file. The numbers must correspond to active subcases. The default for analysis is SUBCASE=all. There is no default for any other procedures, if there are multiple subcases, the subcase for the procedure must be specified.

Examples:

```
subcase = 3,24,36  
sub = 2  
subcase=all
```

Remarks:

1. This directive is not required if there is only one single SUBCASE in the NASTRAN input file.

CONTROL

SUPER

Condensation of Non-active Regions of the Model

SUPER

Elements not defined by the DOMAIN or SDOMAIN collectors are integrated during the very first analysis only.

Nodes not defined by DOMAIN, SDOMAIN, UNLOCKED, OBJECTIVE and CONSTRAINT collectors are condensed.

Examples:

super

Remarks:

1. The SUPER command can be used only for design improvement; i.e. not for ANALYSIS or SENSITIVITY.
2. SUPER is not supported for modal objective or modal constraints.
3. The domain must contain less than 50% of the elements for the entire model, else SUPER will be deactivated.
4. The condensation will be activated only if there is at least one iteration step.
5. The super-element data is written to the file *.sup.
6. If the SUPER command is used, and the correct *.sup file is already available, the super-element will be used for the improvement process, as long as the elements in the condensed region have not been changed. This substantially reduces computing time.
7. Loads, constraints, objective and constraint sets, even nodal co-ordinates (as long as the node numbers are not changed) can be changed in the active region (DOMAIN, SDOMAIN, UNLOCKED, OBJECTIVE and CONSTRAINT) without requiring new condensation.
8. The control points for PROCESS(FUNCTION) can be changed anywhere in the model without requiring new condensation.

RESPONSES

RESPONSES Block

Commands:

COMPLIANCE
DISPLACEMENTS
EXTERNAL
MASS, VOLUME, AREA, LENGTH
MODE
REACTION
STRESS
BUCKLING

The RESPONSES block specifies the objective targets and constrained responses of the selected design improvement process.

Only one RESPONSES block should appear in a command file.

Maximum number of responses is 12.

RESPONSES

Header of RESPONSES Block

RESPONSES(([, STRATEGY = method][, FMIN = fmin][, PRESTRESS = file])

Parameters

Meaning

method

STRATEGY=NEAREST (Default) selects the response with the current value closest to its target value as the objective. All the other responses are considered as constraints with BOUND=nolimit. TARGET values MIN and MAX not allowed.

STRATEGY =FURTHEST) selects the response with the current value furthest from its target value as the objective. All the other responses are considered as constraints with BOUND=nolimit. TARGET values MIN and MAX not allowed.

STRATEGY = sequential will resolve the first objective before moving onto the second objective and so on in the user-defined order.

STRATEGY = Cn. The integer n specifies how many iterations should be applied to the current objective before moving on to the next objective. The responses are selected sequentially in the order given.

fmin

Fmin is the minimum frequency to be used for the response MODE. Default is 0.0 Hz.

file

Path to file containing element shell stresses in NASTRAN punch format for performing modal analysis/improvement with initial stresses.

Examples:

```
resp(strat=f)
resp(strat=c10000) !achieve first objective before going onto second objective
resp(FMIN=1000)
resp(fmin=1000,strat=c1)
responses()
```

RESPONSES

Remarks:

1. Only one RESPONSES block can appear in a ReSHAPE command file.
2. The RESPONSES block is compulsory for sensitivity and improvement processes.
3. After the first objective has reached the target, it is changed to CONSTRAINT with BOUND=hold and the next response is selected as objective in accordance with the strategy.
4. If shift is required for response MODE, the value fmin must be specified here, because EIGRL card is ignored.
5. The prestress file can be supplied in the MSC.Nastran PUNCH format (*.pch file) with stresses exported using the STRESS=ALL command in the SUBCASE block. Alternatively, an engineer may choose to write a program to convert another stress format into the format specified by the following C file output statements:

```
FILE *fw = fopen("filename.pst", "w");
fprintf(fw, "PRESTRESS DATA INPUT 001\n");
fprintf(fw, "%8d\n", elemid);
fprintf(fw, "%+13.6e%+13.6e%+13.6e%+13.6e%+13.6e%+13.6e\n", str[0],
str[1], str[2], str[3], str[4], str[5]);
```

The float array str[] should contain the following (for 3D solids):

str[0] = Stress in the normal-x direction
str[1] = Stress in the normal-y direction
str[2] = Stress in the normal-z direction
str[3] = Stress in the shear-xy direction
str[4] = Stress in the shear-xz direction
str[5] = Stress in the shear-yz direction

Note for beams/bars there will be only str[0] which will be the axial stress. For shells there will only be str[0] (normal-x), str[1] (normal-y) and str[2] (shear-xy).

Example prestress file in this format:

```
PRESTRESS DATA INPUT 001
  745
+8.032200e+00-1.682300e+01+2.212400e+00-4.688800e+00-6.526800e+00+7.393100e+00
  746
+4.707100e+00+3.587400e+01+1.083000e+01-1.019600e+01-5.730600e+00+4.685400e+00
  747
+2.048300e+01+1.069800e+02+9.482800e+00-1.238300e+01-1.561600e+01+4.297100e-01
```

COMPLIANCE

Structural Compliance

COMPLIANCE([SUBCASE = case])

Structural compliance for the specified subcase or default subcase if no subcase is specified. In general terms minimising the compliance will result in a stiffer structure.

Compliance is calculated for the selected subcase as: $C = \mathbf{u}^T \mathbf{K} \mathbf{u}$

Parameters

Meaning

SUBCASE

Subcase that the response applies to. Integer. Only required when more than one subcase is in the input deck.

Examples:

As objective:

compliance(subcase=1, target=min)

comp(targ=min)

DISPLACEMENT

Nodal Displacements

$$\text{DISPLACEMENT} \left(\begin{array}{l}
 \left[\text{, VARIABLE} = \text{var} \right] \left\{ \begin{array}{l} \text{AVGS} \\ \text{AVGW} \end{array} \right\} \left[= \text{level} \right] \\
 \left[\begin{array}{l} \text{TARGET} \\ \text{BOUND} \end{array} \right] = \left\{ \begin{array}{l} \text{targ}[\text{/btype}] \\ \text{btype}[\text{/level}] \end{array} \right\} \\
 \left[\text{, TOLERANCE} = \text{tol} \right] \left[\text{, PMULTIPLIER} = \text{pscale} \right] \\
 \left[\text{, STYPE} = \text{type} \right] \left[\text{, SUBCASE} = \text{case} \right]
 \end{array} \right) = \left\{ \begin{array}{l}
 \text{SET}(\text{setnames}) \\
 \text{PID}(\text{pids}) \\
 \text{ETYPE}(\text{etypes}) \\
 \text{ELEMENT}(\text{eids}) \\
 \text{NODE}(\text{nids})
 \end{array} \right\}$$

Displacement response is calculated in the nodes specified by the collector.

Parameters

Meaning

VARIABLE

The displacement variable is calculated in the local coordinate system, as defined for each node in the analysis (global system if no local system is defined). A single component descriptor only is allowed. For example x-translation [TX] or total rotation [RT] (absolute of X, Y, Z rotations) may be specified. Refer to Table 6 for displacement component codes.

AVGS/ AVGW

Specify averaging on the nodes included in the displacement response set, to determine a single value for the response. AVGS performs a simple average, AVGW performs a weighted average.

Default averaging is AVGW = 90%. The '%' symbol is optional.

The *level* value reduces the set content only to those nodes, at which the value of the variable is above the *level*.

TARGET

The target value if the response is an objective. Can be entered as a specific target value, a percentage (positive or negative) or as either of the keywords 'MIN' or 'MAX'. For example, -10% to decrease the objective value by 10%. For multiple target responses, the type of constraint the target becomes when reached can be optionally specified with the 'btype' parameter. Default is HOLD. See the BOUND option.

BOUND

The *btype* to use if the response is constrained. Can take the values BOUND=CEILING, BOUND=FLOOR BOUND=HOLD, BOUND=NOLIMIT or BOUND=TOPOLOGY.

RESPONSES

The bound types CEILING, FLOOR and HOLD can optionally specify *level* indicating the value of the response at which the constraint becomes active. *Level* can be entered as an absolute value or percentage. The default level is 0%.

BOUND=TOPOLOGY/*level* projects the objective sensitivity vector onto the tangent of the constraint sensitivity vector without checking the scalar value of the constraint – the improvement process will cease when ‘level’ is reached between two consecutive steps (optional). The default of *level* is 100% - i.e., if the constraint doubles between two steps, stop the improvement.

See remarks for more information.

TOLERANCE	Tolerance on a target or constraint value. Default is 1%. Can be entered either as a specific value, or a percentage. When a target value is crossed, CPARAMETER(MAXREDO) defines how many steps ReSHAPE will take trying to iterate back to the target to be within the tolerance.
PMULTIPLIER	Scalar multiplier for the value of TOLERANCE defining the band from a constraint response inside which the objective will be projected onto the constraint. Applies to constraints only. Default is 2. See remark 6.
STYPE	Type of the elements to be selected from the domain for the response, for example STYPE = 2d. This is used only when DOMAIN consists of elements of different types (1d, 2d, 3d). It allows constraints and objectives to be set on different element types (since sensitivities cannot be mixed between elements). If more than one type of element is in the active domain (after LOCKED and PASSIVE elements have been accounted for), then any objective or constraint commands on different element types must specifically say the type of element being used for that constraint or objective using the STYPE parameter. All element types used in the domain must be addressed with STYPE.
SUBCASE	Subcase that the response applies to. Integer. Only required when more than one subcase is in the input deck.
SET(setnames)	Nodes where objective response is to be evaluated, selected in sets. Multiple sets can be defined (Remark 3). setnames (character) contains the identification names of NSETs and ESETs.
PID(pids)	Nodes where objective response is to be evaluated, selected with respect to their property ID (Remarks 3&4). pids is the property identification number (integer > 0). Properties can be defined by single integers (i.e. $i_1, i_2, i_3, \dots, i_n$ where $i_1 < i_2 < i_3 < i_n$) or by a range with the format i_m THRU i_n , where i_m

RESPONSES

is the first property number and i_n the last property number ($i_m < i_n$). Property identification numbers that do not exist are ignored. ALL may be specified for all properties.

ETYPE(etypes) Nodes where objective response is to be evaluated, selected with respect to their element type, or ALL (Remarks 3&4). etypes is the type of element. For example: CQUAD4. Multiple element types can be defined in each ETYPE collector, separated by commas. ALL specifies all element types, 1D specifies all CBEAM, CROD and CBAR types, 2D all CQUAD, CQUAD4, CTRIA and CTRIA3 types, and 3D all CHEXA, CPENTA and CTETRA types.

ELEMENTS(eids) Nodes where objective response is to be evaluated, selected with respect to their element ID or by ALL for nodes on all elements (Remarks 3&4). eids are the element identification numbers (integer > 0). Elements can be defined by single integers (i.e. $i_1, i_2, i_3, \dots, i_n$ where $i_1 < i_2 < i_3 < i_n$) or by a range with the format i_m THRU i_n , where i_m is the first element number and i_n the last element number ($i_m < i_n$). Identification numbers that do not exist are ignored. ALL may also be specified.

NODE(nids) Nodes where objective response is to be evaluated, selected with respect to their node ID. NODE(ALL) is not supported. nids is the node identification number (integer > 0). Nodes can be defined by single integers (i.e. $i_1, i_2, i_3, \dots, i_n$ where $i_1 < i_2 < i_3 < i_n$) or by a range with the format i_m THRU i_n , where i_m is the first node number and i_n the last node identification number ($i_m < i_n$) in a range. Identification numbers that do not exist are ignored.

Examples:

As objective:

```
displacement(var=ty, target=-25%, avgw=95, tol=5.0)=PID(1 THRU 10, 13,50)
displ(var=rz, avgs=75%, targ=min)=set(TOP_SURFACE, BOTTOM_SURFACE)
```

As constraint:

```
displacement(bound=hold, var=ty, avgw=95, tol=5.0)=PID(1 THRU 10, 13,50)
disp(bound=100, var=tt, stype=1d)=etype(1d)
disp(bound=floor/58.5, var=tt, stype=1d)=etype(1d)
disp(bound=nolimit, var=ta, avgw=65)=node (1 thru 1000000)
```

Remarks:

1. The set identification names used to define a displacement response collector can be either an NSET or ESET. When an ESET is defined, the nodes connected to the specified elements are used.
2. AVGS or AVGW is always given as a percentage. '%' sign optional.

RESPONSES

- When a list of ELEMENTS is defined for a displacement response collector, the nodes of the specified elements are used. Refer to Appendix B for the NASTRAN element types appropriate for the response keyword DISPLACEMENT.
- User should take care that all nodes in any set are in the same coordinate system, as used for the analysis.
- A constrained response can be in three modes of activity, controlled by the BOUND keyword:
 - BOUND = HOLD – active all the time, constrained quantity kept at initial value, which is default
 - BOUND = CEILING – active only when at or above the original level.
 - BOUND = FLOOR – active only when at or below the original level.
 - BOUND = NOLIMIT used for watching the constraint. No limit is applied, but the results will be charted (in the PC version) and also printed out in the summary file (*.sum).
- If the objective is outside the band $pscale*tol$, it is scaled down until it is inside the band. It is then projected onto the constraint and iterated along the projection until it satisfies the tolerance value, tol.

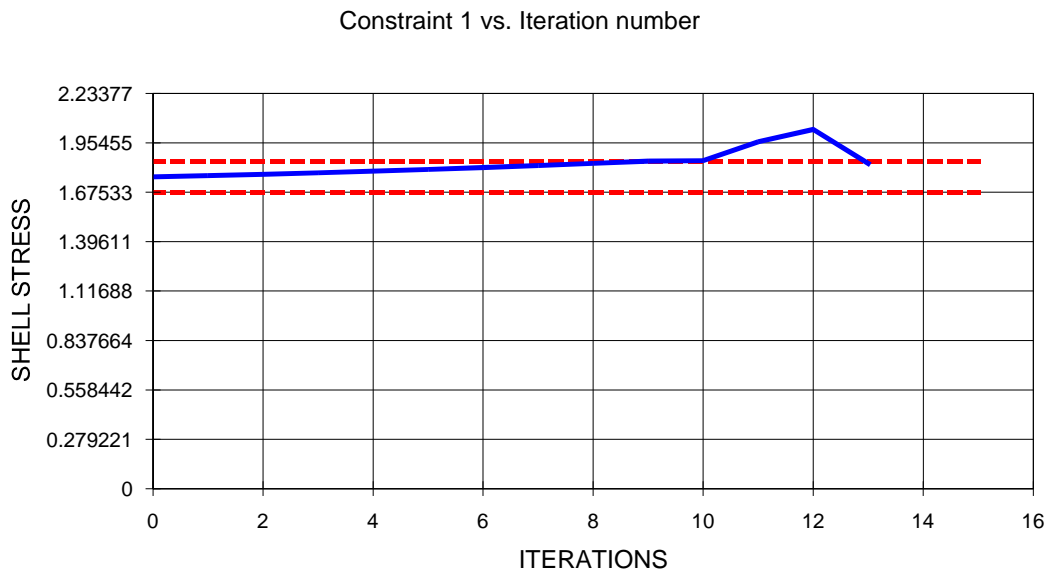


Figure 1 Example of 'BOUND=HOLD' (bound=hold,tol=5%)

RESPONSES

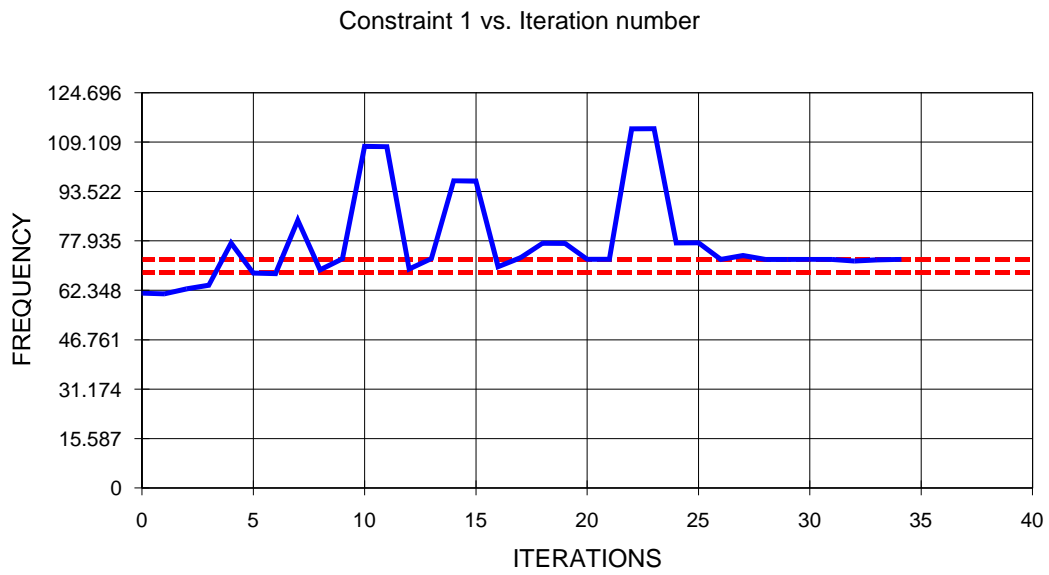


Figure 2 Example of 'BOUND=value' (bound=70,tol=3%)

Table 6 Displacement component descriptor types

Variable	Component
TX	Translation X
TY	Translation Y
TZ	Translation Z
RX	Rotation X
RY	Rotation Y
RZ	Rotation Z
TT	Total Translation
RT	Total Rotation

RESPONSES

EXTERNAL

Read Constraint Data from External File

EXTERNAL(FILE=fname)

Use a file containing sensitivity results as a constraint.

Parameters	Meaning
FILE	Name of the external file containing constraint data in the sensitivity (*.ssh or *.osh) format

Examples:

```
ext(f=fred.ssh)  
external(file=load2-sensitivity.ssh)
```

Remarks:

1. Up to ten (10) EXTERNAL commands can be used.
2. This feature is useful, for example, to achieve an improvement of a component under one load case by constraining it by the previously calculated improvement under a different load case.
3. Another useful application is in the case of large models, when moderate changes in a constraint variable are to be expected. The sensitivity of the constraint is calculated in a separate run, using the sensitivity procedure and the constraint variable as objective, and keeping the *.ssh sensitivities file.
4. If the sensitivity of the *.ssh file is too localized, you can use the *.osh file from PROCESS(RAW) after a sufficient number of iterations instead.

RESPONSES

MASS, VOLUME, AREA, LENGTH

$$\text{MASS} \left(\begin{array}{l} [, \text{TARGET} = \text{targ}] [, \text{BOUND} = \text{btype}] \\ [, \text{STYPE} = \text{type}] [, \text{TOLERANCE} = \text{tol}] \\ [, \text{PMULTIPLIER} = \text{pscale}] \end{array} \right) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Parameters:

See DISPLACEMENT response command

Examples:

As objective:

```
Mass(target=-10%)=set(PART1,PART2)  
vol(target=150)=etype(ctetra)
```

As constraint:

```
area(bound=hold)=pid(9)  
length(bound=ceiling, tol=0.5)=etype(cbeam)
```

Remarks:

1. Refer to Appendix B for the NASTRAN element types appropriate for the response keywords MASS, VOLUME, AREA and LENGTH.

RESPONSES

MODE, BUCKLE

Modal or Buckling Eigenvalue

$$\text{MODE} \left(\begin{array}{l} [\text{VARIABLE} = F_n][, \text{TARGET} = \text{targ}][, \text{BOUND} = \text{btype}] \\ [, \text{TOLERANCE} = \text{tol}][, \text{PMULTIPLIER} = \text{pscale}] \end{array} \right)$$

Parameters

Meaning

VARIABLE

F_n is the frequency (modal) number of the response. For example, if mode 1 or buckle 1 is the response, then VARIABLE=f1.

For other parameters see the DISPLACEMENT command.

Examples:

As objective:

```
mode(var=f1, target=-10%, tol=3%)
```

```
buckle(var=f1, target=10)
```

As constraint:

```
mode(var=f3, bound=max, tol=2%)
```

```
buckle(var=f2, bound=1.0)
```

Remarks:

1. The modal response is calculated for the entire model.
2. EIGRL card is ignored for sensitivity and design improvement.
3. ReSHAPE avoids the rigid body modes in free vibration by using an eigenvalue shift of 10 % of the value fmin of the RESPONSE card.

RESPONSES

REACTION

Reactions in Fixed Nodes

$$\text{REACTION} \left(\begin{array}{l} [, \text{VARIABLE} = \text{var}] \left[\left\{ \begin{array}{l} \text{AVGS} \\ \text{AVGW} \end{array} \right\} [= \text{level}] \right] \\ [, \text{TARGET} = \text{targ}] [, \text{TOLERANCE} = \text{tol}] \\ [, \text{STYPE} = \text{type}] [, \text{BOUND} = \text{btype}] \\ [, \text{PMULTIPLIER} = \text{pscale}] [, \text{SUBCASE} = \text{case}] \end{array} \right) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Parameters

Meaning

VARIABLE

The reaction variable is calculated in the local coordinate system for that node. See Table 7 for variables.

For other parameters see the DISPLACEMENT command.

Examples:

As objective:

```
reaction(var=mt, target= -20%, avgw=95)=set(PLATE_HOLES)
react(target=10, var=fz, ttol=1%, tol=1%)=pid(21)
```

As constraint:

```
reaction(bound=ceiling, var=fy, avgs)=etype(cquad4)
reaction(bound=20, var=fx, avgw=65, stype=1d)=node(1 THRU 1000000)
```

Remarks:

1. The set identification names to define a displacement response collector can be either an NSET or ESET. When an ESET is defined, the nodes connected to the specified elements are used.
2. Default weighting for REACTION is $\text{AVGW} = 90\%$.

RESPONSES

Table 7 Reaction component descriptor types

Variable	Component
FX	Force X
FY	Force Y
FZ	Force Z
MX	Moment X
MY	Moment Y
MZ	Moment Z
FT	Total Force
MT	Total Moment

RESPONSES

STRESS

$$\text{STRESS} \left(\begin{array}{l} [, \text{VARIABLE} = \text{var}] \left[\left\{ \begin{array}{l} \text{AVGS} \\ \text{AVGW} \end{array} \right\} [= \text{level}] \right] \\ [, \text{TARGET} = \text{targ}] [, \text{BOUND} = \text{btype}] \\ [, \text{TOLERANCE} = \text{tol}] [, \text{STYPE} = \text{type}] \\ [, \text{PMULTIPLIER} = \text{pscale}] [, \text{SUBCASE} = \text{case}] \end{array} \right) = \left. \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\} \text{Element Stresses}$$

Parameters

Meaning

VARIABLE Stress variable to be evaluated as the response. Only a single variable is allowed. Refer to Table 8 of stress component codes.

For other parameters see the DISPLACEMENT command.

Examples:

As objective:

```
stress(var=vmmax,targ= -10%,avgw=95,bound=200)=set(FLANGE_ELEMENTS)
```

```
stress(var=fa, target=min)=etype(celas1,celas2)
```

```
stress(var=pmins,avgs,target=min)=pid(200 thru 205)
```

As constraint:

```
stress(bound=ceiling, var=vmmax, tolerance=0.5%)=element (2100 thru 2700)
```

```
stress(bound=350,var=sxxg,stype=3d)=etype(3d)
```

Remarks:

1. Stress responses can only be calculated for element sets consisting only of shells, or only of solids, or only of beams, rods and springs. If PMAXS or VMMAX stress component parameters are used, the user must ensure that the element set specified consists only of shells or only of solids – this is because these parameters are applicable to more than one type of element. All other stress parameters are applicable to only one type of element i.e., only 2D elements. If the collected set includes element types to which the type of stress does not apply, those elements will be ignored.
2. Refer to Appendix B for the NASTRAN element types which are appropriate (in accordance with Table 8) for the response keyword STRESS.
3. Default averaging for a STRESS response is AVGW=90%. The *level* value reduces the set content only to those elements, at which the value of the variable is above the *level*.

RESPONSES

Table 8 Stress component descriptor types

Variable	Component	Available Elements
VMMAX	Maximum von Mises stress	Shells, Solids
VMT	Von Mises stress, top surface	Shells
VMB	Von Mises stress, bottom surface	Shells
PMAXS	Maximum principal stress	Shells, Solids
PINT	Intermediate principal stress	Shells
PMINS	Minimum principal stress	Shells, Solids
PMAXT	Maximum principal stress, top surface	Shells
PMINT	Minimum principal stress, top surface	Shells
PMAXB	Maximum principal stress, bottom surface	Shells
PMINB	Minimum principal stress, bottom surface	Shells
SXXT	Stress in element XX direction, top surface	Shells
SYYT	Stress in element YY direction, top surface	Shells
SXYT	Stress in element XY direction, top surface	Shells
SXXB	Stress in element XX direction, bottom surface	Shells
SYYB	Stress in element YY direction, bottom surface	Shells
SXYB	Stress in element XY direction, bottom surface	Shells
SXXG	Stress in global XX direction	Solids
SYYG	Stress in global YY direction	Solids
SZZG	Stress in global ZZ direction	Solids
SXYG	Stress in global XY direction	Solids
SXZG	Stress in global XZ direction	Solids
SYZG	Stress in global YZ direction	Solids
FMXX	Membrane force XX direction	Shells
FMYY	Membrane force YY direction	Shells
FMXY	Membrane force XY direction	Shells
BMXX	Bending/twisting moment XX direction	Shells
BMYY	Bending/twisting moment YY direction	Shells
BMXY	Bending/twisting moment XY direction	Shells
BMYA	Bending moment around first node, Y axis	Beams, Springs
BMZA	Bending moment around first node, Z axis	Beams, Springs
BMYB	Bending moment around second node, Y axis	Beams, Springs
BMZB	Bending moment around second node, Z axis	Beams, Springs
FA	Axial force	Beams, Rods, Springs
TQ	Torque	Beams, Springs

SIDECONDITIONS

SIDE CONDITIONS block

Commands:

ALIMITS
AVERAGE
MASTERSLAVE
NLIMITS
RLIMITS
TAVERAGE

A maximum of fifteen (15) side condition commands can be specified.

Each command constrains the movement of the structure during the improvement process.

Side conditions are treated differently depending on the process:

PROCESS	AVERAGE	LIMITS	MASTER-SLAVE
RAW	√	√	√
FUNCTION	√	√	CP
GEOMETRY	√	√	√
VECTOR	N/A	√	N/A
SIZING	N/A	N/A	N/A
TOPOLOGY	√ as TAVERAGE	N/A	N/A

√ – before the new analysis

CP – as constraint by generating an internal control point

N/A – not applicable

SIDECONDITIONS

SIDECONDITIONS

Side Conditions Block Header

SIDECONDITIONS

This command is the block header for commands relating to the specification of conditions and constraints on the change of nodal design variable components during shape optimisation.

Examples:

side
sidecon

Note: Side conditions are processed before spline fitting in PROCESS(GEOMETRY) which can make the side conditions behave in an unfamiliar manner.

There is a specific averaging command used for PROCESS(TOPOLOGY) only, called TAVERAGE.

ALIMITS

Absolute Limit on Nodal Design Variables

$$\mathbf{ALIMITS}(\text{component}[= \text{value}] [, \text{component}[= \text{value}], \dots] [, \text{CID} = \text{cid}]) = \left. \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Specifies an absolute limit on a set of nodes, defined as an upper or lower limit. See also RLIMITS for specifying limits relative to the starting configuration.

Parameters	Meaning
component	The component to which a limit is to be defined. Components are interpreted with respect to the coordinate system specified by the CID command. The component consists of the degree of freedom and the direction of the limit – upper or lower. Available components are XU, XL, YU, YL, ZU and ZL. Multiple components may be specified in a single ALIMITS command.
value	The upper or lower limit value. Default is 0.
CID=cid	Coordinate system in which the limit is defined. Default is 0 (global system).

For other parameters see the DISPLACEMENT command.

Examples:

```
alimits(xu=22.5,cid=1)=set(topsurface)
alim(y1,yu=0.5, zu=1.0,cid=5)=node(1 thru 26)
al(y1=0.5,yu=1.5)=etype(cquad4,ctria3)
```

Remarks:

1. The specified coordinates of all nodes in the set must lie within the limits in the starting configuration.
2. In a cylindrical system, $x \Rightarrow r$, $y \Rightarrow \phi$, $z \Rightarrow z$ (NASTRAN definition).
3. In a spherical system, $x \Rightarrow r$, $y \Rightarrow \theta$, $z \Rightarrow \phi$ (NASTRAN definition).

AVERAGE

Averaging of Design Variable Component Changes

$$\mathbf{AVERAGE} \left(\begin{array}{l} [\text{CCOMPONENT} = \text{comp}] [\text{, CCID} = \text{cid}] [\text{, CEPS} = \text{real}] \\ [\text{, ACOMPONENT} = \text{comp}] \end{array} \right) = \left. \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Specifies nodal movement averaging during shape changing process. It defines the basis for a collection of nodes into groups followed by details of averaging to be carried out on each group. The averaging is done in the coordinate system of the DOMAIN.

Parameters	Meaning
CCOMPONENT	The component of nodal design variables to be <u>collected</u> as the basis for grouping or excluding nodes to undergo averaging of their sensitivity values. Nodes having the same spatial coordinate value (within the defined tolerance) prior to the sensitivity calculation belong to this set.
comp	The displacement component identifier X, Y, Z, XY, YZ, ZX, XYZ. The default (blank) is all components; XYZ.
CCID=cid	The coordinate system defining the directions of averaging of the CCOMPONENT group. The default (no CCID specified, or CCID=0) refers to the global coordinate system.
CEPS	The tolerance on component values that are to be regarded as identical, and grouped in a set for averaging. Default is 0.01.
ACOMPONENT	Directional component(s) of nodal sensitivities upon which averaging is to be done. The averaging is done on groups of nodes formed from the CCOMPONENT command.
	For other parameters see the DISPLACEMENT command.

AVERAGE

Examples:

```
aver(ccomp=xy,ccid=1,ceps=0.01,acomp=Z)=set(topsurface)
```

```
ave(ccomp=z,acomp=z)=set(hole2)
```

```
av(ccomp=yz,acomp=z)=node(1 thru 26)
```

```
average(ceps=0.005,acomp=yz)=etype(cquad4,ctria3)
```

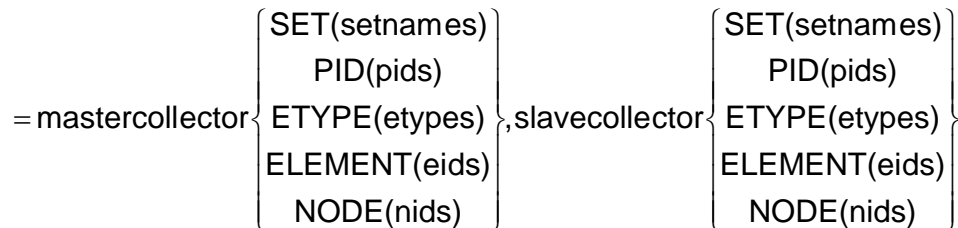
Remarks:

1. The parameters and their values, in the brackets on the left-hand side, as shown above in the standard format, may be omitted. Then the meaning is:
 - a) For each set of nodes having the same local X, Y and Z coordinates within the default tolerance, a group id collected
 - b) For each of these groups, the local X, Y, and Z values are averaged
 - c) These averages are then applied to all nodes in each group.
2. The user has to be careful with using command PASSIVE for the sets to be correctly averaged.
3. In the process GEOMETRY, sensitivities will be calculated in all the nodes of the patch in order to calculate the average motion from the sensitivity of all the nodes.

MASTERSLAVE

Associative Control on Nodal Design Variables

MASTERSLAVE(component [,CID = cid][,EPS = eps])



Defines a set of nodes (slaves) that are changed based on the sensitivity of other nodes (masters). Nodes in a master-slave relationship are also known as independent and dependent respectively.

Parameters

Meaning

component	The component of the slave nodes that is to be influenced by changes to the master nodes. Components are interpreted with respect to the coordinate system specified by CID. Available parameters are X, Y, Z, XY, XZ, YZ and 0 (zero). All nodes with the same co-ordinates as the master will be slaves of that master. Zero is used for the case of one node only being the master of all selected slaves.
CID = cid	Defines the coordinate system for the master-slave relationship.
EPS = eps	The tolerance of grouping slaves to one master. Default is 0.01 length units.
	For other parameters see the DISPLACEMENT command.

Examples:

```

master(x,cid=1)=set(masterset),set(slaveset)
masterslave(yz)=node(2),elem(50,52,54,56,58,60 thru 100)

```

SIDECONDITIONS

NLIMITS

Relative Limit on Nodal Design Variables in Normal Direction

$$\text{NLIMITS}(\text{UPPER}[= \text{value}] [, \text{LOWER}[= \text{value}]) = \left. \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines a relative limit on nodal movement measured in the direction of the normal to the element surface, with respect to the starting configuration. For beams and trusses, there is only one component available (see Remark 2).

Parameters

Meaning

value	The upper or lower limit value. Default is $\pm 1.0\text{E}+20$, resp.
-------	---

For other parameters see the DISPLACEMENT command.

Examples:

```
nlimit(u=2,l=-2)=set(topsurface)  
Nlimit(u=3)=etype(1d)
```

Remarks:

1. The specified coordinates of all nodes in the set must lie within the limits in the starting configuration.
2. For a beam and a truss element, the limit value defines a cylinder around the element; only the upper limit is relevant.
3. For a solid element, only the outer surface normal is calculated.
4. For the movement of a node at the intersection of more elements and/or more element types, limits will be applied to all possible movement.

SIDECONDITIONS

RLIMITS

Relative Limit on Nodal Design Variables

RLIMITS(component[= value] [,component[= value],...[,CID = cid]) = $\left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \\ \text{NODE(nids)} \end{array} \right\}$

Defines a relative limit on nodal movement, with respect to the starting configuration. See also ALIMITS for specifying absolute limits on nodal design variables.

Parameters:

See ALIMITS for information on parameters

Examples:

```
rlimit(xu=22.5,cid=1)=set(topsurface)
rlimit(zl)=set(hole2)
rlim(yl,cid=5)=nodes(1 thru 26)
rl(yu=1.5)=etype(cquad4,ctria3)
```

Remarks:

5. The specified coordinates of all nodes in the set must lie within the limits in the starting configuration.
6. In a cylindrical system, x=>r, y=> ϕ , z=>z (NASTRAN definition).
7. In a spherical system, x=>r, y=> θ , z=> ϕ (NASTRAN definition).

TAVERAGE

Topology Averaging of Design Variable Component Changes

TAVERAGE([COMPONENT = comp][,CID = cid][,EPS = real]) = $\left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$

Specifies averaging of pseudo-volumes during shape changing process. It defines the basis for a collection of elements into groups followed by details of averaging to be carried out on each group.

Parameters	Meaning
COMPONENT	The component of element design variables to be <u>collected</u> as the basis for grouping or excluding elements to undergo averaging of their sensitivity values. Elements with their centres of volume having the same spatial coordinate value (within the defined tolerance) prior to the sensitivity calculation belong to this set.
comp	The displacement component identifier in the specified coordinate system, one of X, Y, Z, XY, YZ, ZX, XYZ. The default (blank) is all components; XYZ.
CID=cid	The coordinate system defining the directions of averaging of the COMPONENT group. The default (no CID specified, or CID=0) refers to the global coordinate system.
EPS	The tolerance on the center of gravity of each element used when selecting the set of elements by component. Default is 0.01.
	For other parameters see the DISPLACEMENT command.

TAVERAGE

Examples:

```
taver(comp=xy,cid=1,eps=0.01)=set(active)
tave(comp=z)=set(hole2)
tav(ccomp=yz)=elem(1 thru 26)
average(ceps=0.005,)=etype(3d)
```

Remarks:

1. The parameters and their values, in the brackets on the left-hand side, as shown above in the standard format, may be omitted. Then the meaning is:
 - d) For each set of elements having the same X, Y and Z coordinates of their centres of volume within the default tolerance, a group id collected
 - e) For each of these groups, the topology pseudo density is averaged
 - f) These averages are then applied to all elements in each group.

PROCESS

PROCESS Block

Commands:

<i>DOMAIN</i>	<i>must consist of ESETs</i>
<i>PASSIVE</i>	<i>must consist of ESETs</i>
<i>LOCKED</i>	
<i>NORMAL</i>	
<i>UNLOCKED</i>	

THIS PROCESS IS USED ONLY FOR THE CALCULATION OF SENSITIVITIES.

The process block without a parameter (PROCESS, or alternatively PROCESS()) can be used for the calculation of sensitivities only. It is intended as a help to the user when he/she has not decided yet on the specific PROCESS.

See PROCESS(RAW) for all the commands.

Only one PROCESS is permitted.

PROCESS(RAW)

PROCESS(RAW) Block

Commands:

<i>DOMAIN</i>	<i>must consist of ESETs</i>
<i>LOCKED</i>	
<i>NORMAL</i>	
<i>PASSIVE</i>	<i>must consist of ESETs</i>
<i>UNLOCKED</i>	

PROCESS(RAW) is useful only for special cases, for example truss structures, frame structures, or selected beam structures.

Only one PROCESS is permitted.

DOMAIN

Specification of Model Regions to be Re-meshed

$$\mathbf{DOMAIN}(\text{component}[, \text{CID} = \text{cid}]) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines the active part of the model to be iteratively changed by nodal movement to achieve the design objective (for SENSITIVITY process, only the sensitivities are calculated without any iteration). See also PASSIVE and LOCKED for de-activation.

Parameters

Meaning

component	The degree of freedom that is to be changed. Single and multiple components are allowed i.e., X-variation [X] or X and Y variation [XY]. Refer to the Table 7. The default is XYZ.
CID	Coordinate system in which the model change should occur. For other parameters see the DISPLACEMENT command

Examples:

```
domain(X)=pid(21)
domain=set(p1,p2,p34)
domain(xz,cid=3)=sets(plate, stiffeners)
```

Remarks:

1. Only one DOMAIN command is allowed.
2. The set identification names can be only of element type (SETS of ESET type, PIDS, ETYPE, ELEMENTS). The nodes connected to the specified elements will be considered free to change with respect to the coordinate system specified. They can be subsequently fixed by the LOCKED command.

Table 9 Component descriptors

Parameter	Component
X	Translation X
Y	Translation Y
Z	Translation Z
XY or YX	Translations X and Y
YZ or ZY	Translations Y and Z
ZX or XZ	Translations Z and X
Any permutation of XYZ	Translations X, Y and Z

PROCESS(RAW)

LOCKED

Disable Change of Nodes

$$\text{LOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Restricts the movement of a set of nodes in the specified degrees of freedom.

Parameters

Meaning

component Restricted degree of freedom of the node(s). Single and multiple components are allowed i.e., X-variation [X] or X and Y variation [XY]. Refer to Table 9. The default is the DOMAIN components.

For other parameters see the DISPLACEMENT command.

Examples:

```
locked(xyz)=set(ALL)
loc(xyz)=etype(cquad4)
lock(xz)=elements (1 thru 1000000)
```

Remarks:

1. More than one LOCKED command can be used.
2. The commands LOCKED, if used, must follow the command DOMAIN, and if used, the PASSIVE commands.
3. The set identification names can be either of NSET or ESET types. When an ESET, the nodes of the defined elements will be 'locked' in the specified degrees of freedom.
4. The nodes specified will be locked in the components specified, or, if not specified, in the components specified in the DOMAIN command.
5. The components not specified in the DOMAIN command will not be locked.

PROCESS(RAW)

NORMAL

Calculate Sensitivity in Normal Directions

$$\text{NORMAL} = \left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Sensitivity is calculated in the nodes of shell elements and in the external nodes of solid elements in normal direction to the surface.

Examples:

Normal=elem(all)
normal=etype(3d)

Remarks:

1. The normal vector in a node is calculated as an average of the normal vectors to all the elements connected to the node.
2. In the case of solid elements, the faces on the surface of the model are treated as if shell elements.
3. The sensitivity vector in the node is projected on the resulting normal vector and only this projected component is resolved back into global components.
4. Shells and solids cannot be mixed in the right hand side element selection set.

PROCESS(RAW)

PASSIVE

Deactivation of Sensitivity Calculation in Some or All nodes

$$\text{PASSIVE}[\underline{\text{ALL}} | \text{INTERNAL ONLY}] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines nodes (by referring to the governing elements) that the sensitivity to the design variable should not be calculated. Nodal movements and smoothing is still allowed to take place in the specified nodes.

Parameters

Meaning

ALL	Default. Sensitivities are not calculated in any nodes of the defined element sets.
INTERNAL ONLY	Sensitivities are not calculated in internal nodes of the defined element sets. For possible ways of defining the element sets see STRESS command.

Examples:

```
passive=set(plate)
pas=pids(21)
passive(internal)=etype(3d)
```

Remarks:

1. PASSIVE(INTERNAL) command is automatically activated when 2DMODEL mode is specified in control block. The PASSIVE command is therefore not required (or allowed) in process block.
2. Sensitivities in the internal nodes are not calculated when in, i.e. command PASSIVE (INTERNAL) is activated.
3. Internal nodes are those that are not surrounded in any relevant direction by elements (an angle of 2π for shells and a spherical angle of 4π for solids). The elements will be re-meshed.
4. As many PASSIVE commands as necessary may be used.
5. The PASSIVE commands (when used) must follow the command DOMAIN and precede any of the LOCKED commands.
6. PASSIVE can only be applied to elements (SET of ESET type, PID, ETYPE, ELEMENTS). The nodes associated with the defined elements, and free to change, will move with respect to the coordinate system specified for that node. They can be subsequently fixed by the LOCKED command.

PROCESS(RAW)

7. The user has to be careful with using command `PASSIVE` for the sets to be correctly averaged (see command `AVERAGE` in the `SIDE CONDITIONS` block).

PROCESS(RAW)

UNLOCKED

Enable Change of Previously LOCKED Nodes

$$\text{UNLOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Reverses a restraint defined by a LOCKED command for a set of nodes.

Parameters

Meaning

component

Degree of freedom to unlock. Single and multiple components are allowed, e.g. X-variation [X] or X and Y variation [XY]. Refer to Table 9. The default is the DOMAIN components.

For other parameters see the DISPLACEMENT command.

Examples:

unlocked(xyz)=set(ALL)

unlock(x)=pids(21)

Remarks:

1. More than one UNLOCKED command can be used.
2. The UNLOCKED commands, if used, must follow any LOCKED command.
3. The nodes defined will be made free in the components specified, or, if not specified, in the components defined in the DOMAIN command.
4. The components not specified in the DOMAIN command will not be made free.
5. The set identification names can be either of NSET or ESET types. When of ESET types, the nodes of specified elements will be made free.

PROCESS(FUNCTION)

PROCESS(FUNCTION) Block

Commands:

DOMAIN *must consist of elements*
PASSIVE *must consist of elements*
LOCKED
UNLOCKED
FPLANE
FCONTROL
FPARAMETER
NORMAL

The PROCESS (FUNCTION) uses influence functions for smoothing.

Only one FUNCTION block is allowed.

Only one PROCESS is permitted.

DOMAIN

Specification of Model Regions to be Re-meshed

$$\text{DOMAIN}(\text{component}[, \text{CID} = \text{cid}]) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines the active part of the model to be iteratively changed by nodal movement to achieve the design objective (for process SENSITIVITY, only the sensitivities are calculated without any iteration). See also PASSIVE and LOCKED for control of individual node states.

Parameters

Meaning

component The components for model change. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. The default is XYZ. Refer to Table 10.

CID Coordinate system in which the model change should occur.

For other parameters see the DISPLACEMENT command

Examples:

```
domain(X,cid=21)=pid(21)
domain(xz)=sets(plate, stiffeners)
```

Remarks:

1. Only one DOMAIN command is allowed.
2. Sets can only consist of elements (SETS of ESET type, PIDS, ETYPE, ELEMENTS). The nodes connected to the specified elements will be considered free to change with respect to the global coordinate system. They can be subsequently fixed by the LOCKED command.
3. For PROCESS(FUNCTION), internal control points are automatically created at the boundary between the domain and the rest of the model.
4. For PROCESS(FUNCTION), only nodes at the free boundary should be locked by the LOCKED command, if required. DO NOT LOCK ANY INTERNAL NODES WITH LOCKED COMMANDS! This could cause excessive requirement on memory size, due to a large number of internal control points created and subsequently eliminated.

PROCESS(FUNCTION)

Table 10 Component Parameters

Describer	Component
X	Translation X
Y	Translation Y
Z	Translation Z
XY or YX	Translations X and Y
YZ or ZY	Translations Y and Z
ZX or XZ	Translations Z and X
Any permutation of XYZ	Translations X, Y and Z

FCONTROL

Control Points and the Domain of their Influence

$$FCONTROL \left([WRT = comp] \left[, ROTATION = \begin{cases} YES \\ NO \end{cases} \right] \right) = \begin{cases} SET(setname) \\ NODES(nids) \\ PID(pids) \\ ETYPES(etypes) \\ AUTO \\ AUTO1 \\ AUTO2 \end{cases}$$

Parameters

WRT=comp	The influence functions are calculated <i>with respect to</i> the component in the CID of the domain. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Default WRT is the same degrees of freedom as specified in the DOMAIN command.
ROTATION	Rotation of control points permitted, creating extra influence functions. Default is NO.
AUTO	If AUTO is specified on the right hand side then one control point is automatically placed.
AUTO1	If AUTO1 is specified on the right hand side then 8 control points are placed at the extremities of the domain in 3D space.
AUTO2	If AUTO2 is specified on the right hand side then 27 control points are placed at the extremities of the domain in 3D space and half-way between the extremities.

For the other parameters see the DISPLACEMENT command.

Examples:

```
fcontrol=nodes(212,451,32)
fc(wrt=xy,r=y) =set(cpoints)
fcontrol=auto2
fcontrol=auto
```

PROCESS(FUNCTION)

Remarks:

1. AUTO can be used only on a domain that is a partial subset of the model – not a domain that is the whole model. The automatic control point is blended to the locked nodes around the domain to generate a smooth shape.
2. Control points can be visualized as a point from which a virtual displacement bubble function emanates outward in a radial direction.
3. If WRT=xy and ROTATION=YES, only the complementary rotation component, i.e. rotation about the z-axis is activated. Similar for yz and zx components.
4. If WRT=x, no rotation component can be activated. ROTATION=YES will result in a FATAL ERROR message. Similar for y and z components
5. The amplitude of the influence function is calculated for each free component of the control point from the sensitivities of all the sensitive nodes within the domain.
6. The method of defining the control points is usually limited to the first two commands (SET or NODES). Since control points are assigned sparingly, rarely is there a situation that the other methods are appropriate; however they are there if required.
7. The boundary between the DOMAIN and the rest of the model is locked internally.

FPARAMETER

Influence Function Parameters

FPARAMETER(parameter) = value

Table 11 FPARAMETER Parameters and their default values

Parameter	Description	Default Value
<i>AUTO</i>	Sets the distance for the automatic control point feature – a multiplier against the FE model object radius. Must be > 0.	1.25
<i>BOUNDARY</i>	The depth of influence of the constrained control points (0.0<BOUNDARY)	1.0
<i>SUPPORT</i>	Level of intersection between two neighbouring control points (0.0<SUPPORT)	1.5
<i>CUTOFF</i>	The value is a multiplier of the minimum support of influence functions. For nodes at a larger distance for the relevant control point, the influence functions will be 0 (zero).	30.0
<i>RADIALCHANGE</i>	If a local cylindrical or spherical coordinate system is used on the domain, then the ortho-normalisation will move nodes located at $r=0$. This is usually not the expected result, so nodes are locked internally in this direction by default. Set to 'YES' to allow nodes at $r=0$ to move.	NO

Examples:

fparam(bound)=0.8
 fpar(support)=0.2
 fpar(cutoff)=20
 fparam(rad)=YES

FPARAMETER

Remarks:

1. FPARAMETER(BOUNDARY) is the most significant tolerance factor. Since influence functions have unlimited support, the insensitive boundaries of the domain must be constrained to zero. This is achieved by a) generating internal control points in all insensitive nodes, b) calculating influence functions in these points and c) subsequently eliminating them by setting the change of the insensitive nodes to zero. If some unexpected mesh changes close to the insensitive nodes are noticed, the tolerance of the BOUNDARY parameter should be changed.
2. For more than one user defined control point, the pseudo-support of the internally generated control points is calculated as the pseudo-support of the user defined control points multiplied by the value the BOUNDARY parameter.
3. In the case of using only one user defined control point, (i.e. a constant influence function) the pseudo-support of the internally generated control points is calculated as the minimum distance between the user defined control point and the closest internal control point multiplied by the value the BOUNDARY parameter. The coarser the mesh, the larger the BOUNDARY tolerance should be to be able to control the nodes closer to the boundary with the internal influence function.
4. In general, the pseudo-support of the influence function is the value of the minimum distance between pairs of user defined control points multiplied by the SUPPORT tolerance. At this distance, the influence function will decay to about 35% of its peak value. Therefore the SUPPORT parameter controls the degree of the interaction between neighboring control points. To avoid the over-influence of control points, located at large distances from other control points, the pseudo-support is “blended” (i.e. it changes over distance). It begins with the value calculated for a given control point and blends into the value of the minimum pseudo-support over the distance of one pseudo-support for user defined control points, and over two-pseudo-supports for fixed (boundary) control points. Note: Boundary control points have usually much smaller pseudo-support then the user defined control points (default is 10%).
5. The CUTOFF parameter makes the values of influence functions equal to zero (0) at all the nodes that are more distant from the relevant control point than the CUTOFF value multiplied by the pseudo-support.

FPLANE

Keep nodes on a plane

$$\text{FPLANE} \left(\begin{array}{l} \text{PLANE} = n1/n2/n3 \text{ [, TOLERANCE} = \text{ptol]} \\ \text{ , IGNORE} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \end{array} \right) = \left\{ \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \\ \text{NODE(nids)} \end{array} \right\}$$

Nodes contained in a set will be kept on a plane. The plane itself cannot move.

Parameters	Meaning
n1/n2/n3	Three nodes defining the plane
ptol	Tolerance of the nodes to being coplanar. Default is 1.0 (in model units).
IGNORE	Whether a warning message for any nodes outside the specified tolerance is issued. (Default = NO, a warning will be issued).

Examples:

```
fplane(plane=100/120/310)=set(pl1)
fplane(p=1/3/9,tol=1.0E-4,ignore=y)=elem(all)
```

Remarks:

1. More than one FPLANE command can be used.
2. The value of ptol is interpreted in the unit length in the model. The default 1.0 is suitable for nodes more than 1 mm apart. If different units are used, or nodes are closer to each other, the users must select their own tolerance.

LOCKED

Disable Change of Nodes

$$\text{LOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Restricts the movement of nodes in the specified degree(s) of freedom.

Parameters**Meaning**

component

The displacement component to be restricted. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 10. The default is the components from the DOMAIN command.

For other parameters see the DISPLACEMENT command.

Examples:

```
locked(xyz)=set(ALL)
loc(xyz)=etype(cquad4)
lock(xz)=elements (1 thru 1000000)
```

Remarks:

1. More than one LOCKED command can be used.
2. Any LOCKED commands, if used, must follow the DOMAIN command and must also follow any PASSIVE commands.
3. The set identification names can be given as NSET or ESET commands. If an ESET is supplied, the nodes of specified elements will be locked.
4. The nodes will be locked in the specified components, or in the components defined in the DOMAIN command (Default).
5. The components not defined in the DOMAIN command will not be locked.
6. For PROCESS(FUNCTION), the LOCKED command has the additional function of generating internal control points.

PROCESS(FUNCTION)

NORMAL

Calculate Sensitivity in Normal Directions

$$\text{NORMAL} = \left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Sensitivity is calculated in the nodes of shell elements and in the external nodes of solid elements in normal direction to the surface.

Examples:

Normal=elem(all)
normal=etype(3d)

Remarks:

1. The normal vector in a node is calculated as an average of the normal vectors to all the elements connected to the node.
2. In the case of solid elements, the faces on the surface of the model are treated as if shell elements.
3. The sensitivity vector in the node is projected on the resulting normal vector and only this projected component is resolved back into global components.
4. Shells and solids cannot be mixed in the right hand side element selection set.

PASSIVE

Deactivation of Sensitivity Calculation in Some or All Nodes

$$\text{PASSIVE}([\underline{\text{ALL}} | \text{INTERNAL ONLY}]) = \left\{ \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Defines nodes (by referring to the governing elements) that the sensitivity to the design variable should not be calculated. Nodal movements and smoothing is still allowed to take place in the specified nodes.

Parameters	Meaning
ALL	Default. Sensitivities are not calculated in any nodes of the defined element sets.
INTERNAL ONLY	Sensitivities are not calculated in internal nodes of the defined element sets.

For possible ways of defining the element sets see STRESS command.

Examples:

```
passive=set(plate)
pas=pids(21)
passive(internal)=etype(3d)
```

Remarks:

1. PASSIVE(INTERNAL) command is automatically activated when 2DMODEL mode is specified in control block. The PASSIVE command is therefore not required (or allowed) in process block.
2. Internal nodes are those that are not surrounded in any relevant direction by elements (an angle of 2π for shells and a spherical angle of 4π for solids). The elements will be re-meshed.
3. As many PASSIVE commands as necessary may be used.
4. The PASSIVE commands (when used) must follow the command DOMAIN and precede any of the LOCKED commands.
5. PASSIVE can only be applied to elements (SET of ESET type, PID, ETYPE, ELEMENTS). The nodes associated with the defined elements, and free to change, will move with respect to the coordinate system specified for that node. They can be subsequently fixed by the LOCKED command.
6. The user has to be careful with using command PASSIVE for the sets to be correctly averaged (see command AVERAGE in the SIDE CONDITIONS block).

PROCESS(FUNCTION)

UNLOCKED

Enable Change of Previously LOCKED Nodes

$$\text{UNLOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Reverses a restraint defined by a LOCKED command for a set of nodes.

Parameters

Meaning

component

The displacement component to remove restriction from. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 10. The default is the components in the DOMAIN command.

For the other parameters see the DISPLACEMENT command.

Examples:

unlocked(xyz)=set(ALL)

unlock(x)=pids(21)

Remarks:

- 1) More than one UNLOCKED command can be used.
- 2) Any UNLOCKED commands, if used, must follow any LOCKED commands.
- 3) The nodes defined will be made free in the components specified, or in the components specified in the DOMAIN command if no specific components are defined in the UNLOCKED command (Default).
- 4) If only the default UNLOCKED is used, the components not specified in the DOMAIN command will not be made free.
- 5) The set identification names can be input as either an NSET or an ESET. When an ESET is supplied, the nodes of specified element set will be made free.

PROCESS(GEOMETRY)

PROCESS(GEOMETRY) Block

Commands:

DOMAIN *must consist of elements*

PASSIVE *must consist of elements*

LOCKED

NORMAL

UNLOCKED

GCORNERS

GLINE

GRIBS

GTOLERANCE

PROCESS(GEOMETRY) uses cubic splines for smooth shape changes. The splines are pre-calculated by executing the command GSET in the RESHAPE SETS block.

Surfaces are approximated by F-patches in the ReSHAPE (zero twist vectors), therefore sensitivities are not calculated in internal nodes of each patch.

If command AVERAGE is used, sensitivities are calculated in all nodes of the relevant patch (see command AVERAGE in the SIDE CONDITIONS block).

Solids cannot be reshaped by GEOMETRY process.

Only one PROCESS is permitted.

DOMAIN

Specification of Model Regions to be Re-meshed

$$\text{DOMAIN}(\text{component}[, \text{CID} = \text{cid}]) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines the active part of the model to be iteratively changed by nodal movement to achieve the design objective (for process SENSITIVITY, only the sensitivities are calculated without any iteration). See also PASSIVE and LOCKED for control of individual node states.

Parameters	Meaning
component	The displacement component for alteration. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. The default is XYZ. Refer to Table 12.
CID	Coordinate system in which the model change should occur. For information on other parameters see the STRESS command

Examples:

domain(X,cid=21)=pid(21)
domain(xz)=sets(plate, stiffeners)

Remarks:

1. Only one DOMAIN command is allowed.
2. Sets can only consist of elements (SETS of ESET type, PIDS, ETYPE, ELEMENTS). The nodes connected to the specified elements will be considered free to change with respect to the global coordinate system. They can be subsequently fixed by the LOCKED command.
3. Domain for PROCESS(GEOMETRY) must contain at least one GSET.

Table 12 Component Parameters

Describer	Component
X	Translation X
Y	Translation Y
Z	Translation Z
XY or YX	Translations X and Y
YZ or ZY	Translations Y and Z
ZX or XZ	Translations Z and X
Any permutation of XYZ	Translations X, Y and Z

GCORNERS

Specify Corners of Geometric Sets

GCORNERS(SID) = {NODES}

Specifies the nodes that are to be regarded as corners on a geometric region for the purpose of spline formulation. This command is only required when the corners of the region are not distinct from other nodes in terms of the degree of local curvature. See also GPARAMETER.

<i>Parameters</i>	<i>Meaning</i>
SID	The set identification number (integer > 0). Set must be defined by a single integer, and must have been previously been declared as a geometry set using the GSET command.
NODES	The corner nodes of the geometry set (integer > 0). Sets must be defined by a list of integers corresponding to the corners of a geometric spline region. The nodes may be specified in any order, they do not have to be in ascending order.

Examples:

```
gcorners(12)=1,3,5,7
gco(56)=7,12,43
```

Remarks:

1. Line geometry must have two ends/corners, surface geometry may have either three or four corners.

GLINE

Specify a Line that Must Stay Straight

GLINE(*NODES* = *n1/n2* [*TOLERANCE* = *tol*])

Specifies the two end nodes of a line of nodes that must be kept straight. The nodes must be the corners of patches.

Parameters	Meaning
NODES	<i>n1,n2</i> are the end nodes of the line of nodes to keep straight.
TOLERANCE	All nodes that lie in the tolerance band <u>tol</u> from the straight line between nodes <i>n1</i> , <i>n2</i> will be selected and kept on that line (default =1).

Examples:

gline(node=2/20)

Remarks:

1. In order to keep a line of nodes straight, it may be necessary to lock (or make passive) all nodes on the line except for the end nodes, so that the middle nodes do not have any sensitivity.

PROCESS(GEOMETRY)

GPARAMETER

Control Parameters for PROCESS(GEOMETRY)

GPARAMETER(parameter) = *value*

Sets parameters specific to a PROCESS(GEOMETRY) iterative design improvement.

Table 13 GPARAMETER parameters and values

Parameter	Description	Default Value
<i>ANGLE</i>	Defines the angle which defines a geometric 'corner' if GCORNER is not used. If the angle between two sides is less than angle, then ReSHAPE assumes it is a corner.	150°
<i>SMOOTH</i>	The tolerance on smoothing. The upper bound of the angular deviation in the end-nodes of the splines between two adjacent edges in the original model. If the deviation is below the value, the transition is regarded as originally smooth, and is kept smooth.	2°

Examples:

gparam(angle) = 160
gparameter(smooth)=4

Remarks:

1. Using the parameter ANGLE, allows for an easier general definition of corner nodes than can be achieved with the GCORNERS command.
2. The value of SMOOTH must be kept less than 10°, or a warning is issued.

GRIB

Specify a Rib that Must Stay Straight

$$\text{GRIB}\left(\text{GROUP}\begin{Bmatrix} \text{A} \\ \text{B} \end{Bmatrix}, [\text{TOLERANCE} = \text{tol}]\right) = \text{NODES}(\text{n1}, \text{n2}), \text{NODES}(\text{n3}, \text{n4})$$

Specifies top and bottom straight lines of a rib, modeled by shells meshed on multiple patches.

Parameters

Meaning

GROUP	ribs are to be divided in two groups: A and B. Each group consists of mutually non-crossing ribs. One group may or may not cross the other group.
TOLERANCE=tol	tol is the tolerance on the nodes along rib edges to be considered as located on the straight line between the pair of end nodes n1,n2 or n3,n4.
n1, n2	The end nodes of the top boundary line of a patch to be kept straight.
n3, n4	The end nodes of the bottom boundary line of a patch to be kept straight.

Examples:

```
grib(g=a)=nodes(2,20),nodes(4,43)
grib(g=b)=n(1,5),n(34,20)
```

Remarks:

1. The ribs must be embedded between boundary ribs.
2. The end nodes of the boundary patches must be LOCKED.
3. There must be at least one rib of each type (A and B).
4. Some end nodes of ribs may be LOCKED.

LOCKED

Disable Change of Nodes

$$\text{LOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Restricts the movement of nodes in the specified degree(s) of freedom.

Parameters**Meaning**

component

The displacement component to be restricted. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 12. The default is the components from the DOMAIN command.

For other parameters see the DISPLACEMENT command.

Examples:

```
locked(xyz)=set(ALL)
loc(xyz)=etype(cquad4)
lock(xz)=elements (1 thru 1000000)
```

Remarks:

1. More than one LOCKED command can be used.
2. Any LOCKED commands, if used, must follow the DOMAIN command and must also follow any PASSIVE commands.
3. The set identification names can be given as NSET or ESET commands. If an ESET is supplied, the nodes of specified elements will be locked.
4. The nodes will be locked in the specified components, or in the components defined in the DOMAIN command (Default).
5. The components not defined in the DOMAIN command will not be locked.

NORMAL

Calculate Sensitivity in Normal Directions

$$\text{NORMAL} = \left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Sensitivity is calculated in the nodes of shell elements and in the external nodes of solid elements in normal direction to the surface.

Examples:

Normal=elem(all)
normal=etype(3d)

Remarks:

1. The normal vector in a node is calculated as an average of the normal vectors to all the elements connected to the node.
2. In the case of solid elements, the faces on the surface of the model are treated as if shell elements.
3. The sensitivity vector in the node is projected on the resulting normal vector and only this projected component is resolved back into global components.
4. Shells and solids cannot be mixed in the right hand side element selection set.

PASSIVE

Deactivation of Sensitivity Calculation in Some or All Nodes

$$\text{PASSIVE}([\underline{\text{ALL}} | \text{INTERNAL ONLY}]) = \left\{ \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Defines nodes (by referring to the governing elements) that the sensitivity to the design variable should not be calculated. Nodal movements and smoothing is still allowed to take place in the specified nodes.

Parameters	Meaning
ALL	Default. Sensitivities are not calculated in any nodes of the collector.
INTERNAL ONLY	Sensitivities are not calculated in internal nodes of the collector. For possible ways of defining the element sets see STRESS command.

Examples:

```
passive=set(plate)
pas=pids(21)
passive(internal)=etype(3d)
```

Remarks:

1. PASSIVE(INTERNAL) command is automatically activated when 2DMODEL mode is specified in control block. The PASSIVE command is therefore not required (or allowed) in process block.
2. Internal nodes are those that are not surrounded in any relevant direction by elements (an angle of 2π for shells and a spherical angle of 4π for solids). The elements will be re-meshed.
3. As many PASSIVE commands as necessary may be used.
4. For patches not contained in any PASSIVE command, sensitivities are not calculated in the internal nodes of individual patches; i.e. all edges remain sensitive.
5. If PASSIVE(INTERNAL) is used, sensitivities are not calculated in the internal nodes of the collector (internal edges of the collector will be insensitive).
6. The PASSIVE commands (when used) must follow the command DOMAIN and precede any of the LOCKED commands.

PROCESS(GEOMETRY)

7. PASSIVE can only be applied to elements (SET of ESET type, PID, ETYPE, ELEMENTS). The nodes associated with the defined elements, and free to change, will move with respect to the coordinate system specified for that node. They can be subsequently fixed by the LOCKED command.
8. The user has to be careful with using command PASSIVE for the sets to be correctly averaged (see command AVERAGE in the SIDE CONDITIONS block).

UNLOCKED

Enable Change of Previously LOCKED Nodes

$$\text{UNLOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Reverses a restraint defined by a LOCKED command for a set of nodes.

Parameters**Meaning**

component

The displacement component to remove restriction from. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 12. The default is the components in the DOMAIN command.

For the other parameters see the DISPLACEMENT command.

Examples:

```
unlocked(xyz)=set(ALL)
```

```
unlock(x)=pids(21)
```

Remarks:

1. More than one UNLOCKED command can be used.
2. Any UNLOCKED commands, if used, must follow any LOCKED commands.
3. The nodes defined will be made free in the components specified, or in the components specified in the DOMAIN command if no specific components are defined in the UNLOCKED command (Default).
4. If only the default UNLOCKED is used, the components not specified in the DOMAIN command will not be made free.
5. The set identification names can be input as either an NSET or an ESET. When an ESET is supplied, the nodes of specified element set will be made free.

PROCESS(VECTOR)

PROCESS(VECTOR) Block

Commands:

<i>DOMAIN</i>	<i>must consist of elements</i>
<i>LOCKED</i>	
<i>NORMAL</i>	
<i>PASSIVE</i>	<i>must consist of elements</i>
<i>UNLOCKED</i>	
<i>VCREATE</i>	
<i>VINPUT</i>	

The PROCESS (VECTOR) block uses basis vectors as influence functions, either pre-calculated by use of the VCREATE command, or provided by the user in the specified format.

VINPUT commands are used to call the basis vectors in an IMPROVEMENT run.

The constraints of the active domain can be automatically assigned using RESHAPE PROCESS commands (DOMAIIN, LOCKED, UNLOCKED), or taken from the NASTRAN deck by reading the SPC commands.

Multiple VINPUT commands may be used in one IMPROVEMENT run.

Only one PROCESS is permitted.

PROCESS(VECTOR)

DOMAIN

Specification of Model Regions to be Re-Meshed

$$\text{DOMAIN}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Defines the active part of the model to be iteratively changed by nodal movement to achieve the design objective (for PROCESS(SENSITIVITY), only the sensitivities are calculated without any iteration). See also PASSIVE and LOCKED for control of individual node states.

Parameters

Meaning

component

The displacement component of the nodes that is permitted to be changed. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 14. The default is XYZ.

For the other parameters see the STRESS command

Examples:

domain(x)=pid(21)

domain=set(p1,p2,p34)

domain(xz)=sets(plate, stiffeners)

Remarks:

1. Only one DOMAIN command is allowed.
2. If the domain contains only a part of the model, the nodes on the boundary of the domain will be LOCKED.

Table 14 Component Parameters

Describer	Component
X	Translation X
Y	Translation Y
Z	Translation Z
XY or YX	Translations X and Y
YZ or ZY	Translations Y and Z
ZX or XZ	Translations Z and X
Any permutation of XYZ	Translations X, Y and Z

LOCKED

Disable Change of Nodes

$$\text{LOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Restricts the movement of nodes in the specified degree(s) of freedom.

Parameters**Meaning**

component

The displacement component to be restricted. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 14. The default is the components from the DOMAIN command.

For other parameters see the DISPLACEMENT command.

Examples:

```
locked(xyz)=set(ALL)
loc(xyz)=etype(cquad4)
lock(xz)=elements (1 thru 1000000)
```

Remarks:

1. More than one LOCKED command can be used.
2. Any LOCKED commands, if used, must follow the DOMAIN command and must also follow any PASSIVE commands.
3. The set identification names can be given as NSET or ESET commands. If an ESET is supplied, the nodes of specified elements will be locked.
4. The nodes will be locked in the specified components, or in the components defined in the DOMAIN command (Default).
5. The components not defined in the DOMAIN command will not be locked.

PROCESS(VECTOR)

NORMAL

Calculate Sensitivity in Normal Directions

$$\text{NORMAL} = \left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Sensitivity is calculated in the nodes of shell elements and in the external nodes of solid elements in normal direction to the surface.

Examples:

Normal=elem(all)
normal=etype(3d)

Remarks:

1. The normal vector in a node is calculated as an average of the normal vectors to all the elements connected to the node.
2. In the case of solid elements, the faces on the surface of the model are treated as if shell elements.
3. The sensitivity vector in the node is projected on the resulting normal vector and only this projected component is resolved back into global components.
4. Shells and solids cannot be mixed in the right hand side element selection set.

PROCESS(VECTOR)

PASSIVE

Deactivation of Sensitivity Calculation in Some or All Nodes

$$\text{PASSIVE}[\underline{\text{ALL}} \mid \text{INTERNAL ONLY}] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines nodes (by referring to the governing elements) that the sensitivity to the design variable should not be calculated. Nodal movements and smoothing is still allowed to take place in the specified nodes.

Parameters

Meaning

ALL	Default. Sensitivities are not calculated in any nodes of the defined element sets.
INTERNAL ONLY	Sensitivities are not calculated in internal nodes of the defined element sets.

For possible ways of defining the element sets see STRESS command.

Examples:

```
passive=set(plate)
pas=pids(21)
passive(internal)=etype(3d)
```

Remarks:

1. PASSIVE(INTERNAL) command is automatically activated when 2DMODEL mode is specified in control block. The PASSIVE command is therefore not required (or allowed) in process block.
2. Internal nodes are those that are not surrounded in any relevant direction by elements (an angle of 2π for shells and a spherical angle of 4π for solids). The elements will be re-meshed.
3. As many PASSIVE commands as necessary may be used.
4. The PASSIVE commands (when used) must follow the command DOMAIN and precede any of the LOCKED commands.
5. PASSIVE can only be applied to elements (SET of ESET type, PID, ETYPE, ELEMENTS). The nodes associated with the defined elements, and free to change, will move with respect to the coordinate system specified for that node. They can be subsequently fixed by the LOCKED command.
6. The user has to be careful with using command PASSIVE for the sets to be correctly averaged (see command AVERAGE in the SIDE CONDITIONS block).

PROCESS(VECTOR)

UNLOCKED

Enable Change of Previously LOCKED Nodes

$$\text{UNLOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Reverses a restraint defined by a LOCKED command for a set of nodes.

Parameters

Meaning

component

The displacement component to remove restriction from. Single and multiple components are allowed, e.g. X-translation [X] or X and Y translation [XY]. Refer to Table 14. The default is the components in the DOMAIN command.

For the other parameters see the DISPLACEMENT command.

Examples:

```
unlocked(xyz)=set(ALL)
unlock(x)=pids(21)
```

Remarks:

1. More than one UNLOCKED command can be used.
2. Any UNLOCKED commands, if used, must follow any LOCKED commands.
3. The nodes defined will be made free in the components specified, or in the components specified in the DOMAIN command if no specific components are defined in the UNLOCKED command (Default).
4. If only the default UNLOCKED is used, the components not specified in the DOMAIN command will not be made free.
5. The set identification names can be input as either an NSET or an ESET. When an ESET is supplied, the nodes of specified element set will be made free.

PROCESS(VECTOR)

VCREATE

Create File of Shape Basis Vectors

$$\text{VCREATE} \left(\left[\text{FILE} = \text{filename} \right] \left[\text{SPC} = \left\{ \begin{array}{l} \text{INPUT} \\ \text{DOMAIN} \end{array} \right\} \right] \right)$$

VCREATE creates a file of basis vectors. The type of basis vector is dictated by the ANALYSIS command. STATIC creates *displacement vectors*, FREQUENCY *vibration modes*, and BUCKLING *buckling modes*. MUST BE DEFINED BEFORE THE DOMAIN CARD.

Parameters	Meaning
FILE	Name of file for shape basis vectors to be stored in. If the filename descriptor is not used, the filename will be set to <i>filename.vec</i> where <i>filename</i> is the filename of the <i>cmf</i> file used for the run.
SPC	Identifies how the constraints at the edges of the domain are to be determined. If set to DOMAIN, the specific point constraints (SPC's) will be automatically assigned by ReSHAPE in accordance with the DOMAIN, LOCKED and other PROCESS commands or the CONTROL command 2DMODEL. If set to INPUT, the SPC's will be retrieved from the input file for the specified SUBCASE.

Examples

```
vcreate  
vcr(file=dispx.vec,spc=input)  
vcreate(f=modal,spc=dom)
```

Remarks:

1. Basis vectors created by VCREATE command are subsequently used in an improvement process, called with VINPUT commands.
2. VCREATE can only be used once in each RESHAPE(ANALYSIS) run, and must appear before the DOMAIN command. If the analysis is for multiple subcases, all analysed subcases will create a vector in the vector output file.
3. ANALYSIS(VIBRATION) and ANALYSIS(BUCKLING) saves up to 8 vectors in one basis vector file. The number of vectors to be created is dictated by the ANALYSIS command in the CONTROL block.

PROCESS(VECTOR)

VINPUT

Read Basis Vectors from NASTRAN Input File

$$\text{VINPUT} \left[\left(\text{FILE} = \text{filename}, \text{LABEL} = \left\{ \begin{array}{l} \text{ReSHAPE} \\ \text{GRID} \\ \text{DVGRID} \\ \text{FORCE} \end{array} \right\}, \text{VID} = \text{vid} \right) \right]$$

Reads in a vector of nodal changes as a basis for a shape change. VINPUT must be defined before the domain card.

Parameters

Meaning

filename

The file in which the basis vector(s) are stored.

LABEL

The label which refers to the nodal changes. (Default is 'ReSHAPE', which is the label for the output of the basis vectors if VCREATE is used). Other options are 'GRID', 'DVGRID' or 'FORCE'. These can be read from any input file such as those created using the AutoDV template in HyperMesh. Note: the option GRID is used when AutoDV creates \$GRID cards.

vid

The vector identification number, used to select basis vectors from a modal or static analysis vector file created using VCREATE. (Default: VID = ALL)

Examples:

```
vinput(file=disp1.vec,label=ReSHAPE,vid=1thru5,8)
vin(f=vec1.vec)
```

Remarks:

1. Multiple VINPUT commands may be used
2. Basis vector files may be created in any program, such as AutoDV, as long as they are in the NASTRAN 'FORCE' format, and use one of the three available labels.
3. Both VINPUT and VCREATE commands must appear before the DOMAIN card.

PROCESS(VECTOR)

4. AutoDV in HyperMesh creates \$GRID cards for basis vectors. Since the \$ symbol is a comment in ReSHAPE command files, these vectors can be used using the 'GRID' label in the VINPUT command.
5. Multiple basis vectors may be stored in one file, and then called using the VID numbers of the vectors.

PROCESS(EVOLUTION)

PROCESS(EVOLUTION) Block

Commands:

DOMAIN *must consist of ESETs*
LOCKED
NORMAL
PASSIVE *must consist of ESETs*
UNLOCKED
EPARAMETER

PROCESS(EVOLUTION) is a shape change process that can be used for all models.

Only one PROCESS is permitted.

DOMAIN

Specification of Model Regions to be Re-meshed

$$\mathbf{DOMAIN}(\text{component}[, \text{CID} = \text{cid}]) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines the active part of the model to be iteratively changed by nodal movement to achieve the design objective (for SENSITIVITY process, only the sensitivities are calculated without any iteration). See also PASSIVE and LOCKED for de-activation.

Parameters

Meaning

component	The degree of freedom that is to be changed. Single and multiple components are allowed i.e., X-variation [X] or X and Y variation [XY]. Refer to the Table 15. The default is XYZ.
CID	Coordinate system in which the model change should occur. For other parameters see the DISPLACEMENT command

Examples:

```
domain(X)=pid(21)
domain=set(p1,p2,p34)
domain(xz,cid=3)=sets(plate, stiffeners)
```

Remarks:

- Only one DOMAIN command is allowed.
- The set identification names can be only of element type (SETS of ESET type, PIDS, ETYPE, ELEMENTS). The nodes connected to the specified elements will be considered free to change with respect to the coordinate system specified. They can be subsequently fixed by the LOCKED command.

Table 15 Component descriptors

Parameter	Component
X	Translation X
Y	Translation Y
Z	Translation Z
XY or YX	Translations X and Y
YZ or ZY	Translations Y and Z
ZX or XZ	Translations Z and X
Any permutation of XYZ	Translations X, Y and Z

LOCKED

Disable Change of Nodes

$$\text{LOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Restricts the movement of a set of nodes in the specified degrees of freedom.

Parameters**Meaning**

component Restricted degree of freedom of the node(s). Single and multiple components are allowed i.e., X-variation [X] or X and Y variation [XY]. Refer to Table 9. The default is the DOMAIN components.

For other parameters see the DISPLACEMENT command.

Examples:

```
locked(xyz)=set(ALL)
loc(xyz)=etype(cquad4)
lock(xz)=elements (1 thru 1000000)
```

Remarks:

6. More than one LOCKED command can be used.
7. The commands LOCKED, if used, must follow the command DOMAIN, and if used, the PASSIVE commands.
8. The set identification names can be either of NSET or ESET types. When an ESET, the nodes of the defined elements will be 'locked' in the specified degrees of freedom.
9. The nodes specified will be locked in the components specified, or, if not specified, in the components specified in the DOMAIN command.
10. The components not specified in the DOMAIN command will not be locked.

NORMAL

Calculate Sensitivity in Normal Directions

$$\text{NORMAL} = \left. \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Sensitivity is calculated in the nodes of shell elements and in the external nodes of solid elements in normal direction to the surface.

Examples:

Normal=elem(all)
normal=etype(3d)

Remarks:

1. The normal vector in a node is calculated as an average of the normal vectors to all the elements connected to the node.
2. In the case of solid elements, the faces on the surface of the model are treated as if shell elements.
3. The sensitivity vector in the node is projected on the resulting normal vector and only this projected component is resolved back into global components.
4. Shells and solids cannot be mixed in the right hand side element selection set.

PASSIVE

Deactivation of Sensitivity Calculation in Some or All nodes

$$\text{PASSIVE}[\underline{\text{ALL}} \mid \text{INTERNAL ONLY}] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines nodes (by referring to the governing elements) that the sensitivity to the design variable should not be calculated. Nodal movements and smoothing is still allowed to take place in the specified nodes.

Parameters

Meaning

ALL	Default. Sensitivities are not calculated in any nodes of the defined element sets.
INTERNAL ONLY	Sensitivities are not calculated in internal nodes of the defined element sets. For possible ways of defining the element sets see STRESS command.

Examples:

```
passive=set(plate)
pas=pids(21)
passive(internal)=etype(3d)
```

Remarks:

- PASSIVE(INTERNAL) command is automatically activated when 2DMODEL mode is specified in control block. The PASSIVE command is therefore not required (or allowed) in process block.
- Sensitivities in the internal nodes are not calculated when in, i.e. command PASSIVE (INTERNAL) is activated.
- Internal nodes are those that are not surrounded in any relevant direction by elements (an angle of 2π for shells and a spherical angle of 4π for solids). The elements will be re-meshed.
- As many PASSIVE commands as necessary may be used.
- The PASSIVE commands (when used) must follow the command DOMAIN and precede any of the LOCKED commands.
- PASSIVE can only be applied to elements (SET of ESET type, PID, ETYPE, ELEMENTS). The nodes associated with the defined elements, and free to change, will move with respect to the coordinate system specified for that node. They can be subsequently fixed by the LOCKED command.

PROCESS(EVOLUTION)

14. The user has to be careful with using command `PASSIVE` for the sets to be correctly averaged (see command `AVERAGE` in the `SIDE CONDITIONS` block).

PROCESS(EVOLUTION)

UNLOCKED

Enable Change of Previously LOCKED Nodes

$$\text{UNLOCKED}[(\text{component})] = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \\ \text{NODE}(\text{nids}) \end{array} \right\}$$

Reverses a restraint defined by a LOCKED command for a set of nodes.

Parameters

Meaning

component

Degree of freedom to unlock. Single and multiple components are allowed, e.g. X-variation [X] or X and Y variation [XY]. Refer to Table 9. The default is the DOMAIN components.

For other parameters see the DISPLACEMENT command.

Examples:

unlocked(xyz)=set(ALL)

unlock(x)=pids(21)

Remarks:

6. More than one UNLOCKED command can be used.
7. The UNLOCKED commands, if used, must follow any LOCKED command.
8. The nodes defined will be made free in the components specified, or, if not specified, in the components defined in the DOMAIN command.
9. The components not specified in the DOMAIN command will not be made free.
10. The set identification names can be either of NSET or ESET types. When of ESET types, the nodes of specified elements will be made free.

PROCESS(EVOLUTION)

PROCESS(EVOLUTION)

EPARAMETER

Modify sizing parameters

EPARAMETER(parameter) = *value*

Sets parameters specific to a PROCESS(EVOLUTION) iterative design improvement.

Table 16 EPARAMETER parameters and their default values

Parameter	Description	Default Value
<i>INREL</i>	Activates inertia relief for the internal analysis to calculate shape change parameters	NO

Examples:

eparam(inrel)=yes

Remarks:

1. EPARAMETER(INREL) =YES activates the inertia relief method for the internal analysis, when the structure is loaded by automatically generated forces.
2. The use of YES value for this parameter is essential if the structure is only partly constrained by the use of the LOCKED command or by DOMAIN boundary.

PROCESS(SIZING)

PROCESS(SIZING) Block

Commands:

SDOMAIN *must consist of elements*

PROCESS(SIZING) is available for shell elements only.

Only one PROCESS is permitted.

The default stepsize in the STEPS(number, stepsize, multiplier) command in the CONTROL block is calculated as $1.1 * (\text{maxt} - \text{mint}) / \text{number}$, where maxt-mint is the largest difference of these parameters from all sdomain commands.

SDOMAIN

Specification of Model Regions to be Sized

$$\text{SDOMAIN}([\text{MAXT}=\text{maxt}][, \text{MINT}=\text{mint}][, \text{CONSTANTCHANGE}]) = \left\{ \begin{array}{l} \text{SET}(\text{setnames}) \\ \text{PID}(\text{pids}) \\ \text{ETYPE}(\text{etypes}) \\ \text{ELEMENT}(\text{eids}) \end{array} \right\}$$

Defines the active region for possible changes to shell thickness in the iterative design improvement process.

Parameters	Meaning
MAXT=maxt	Upper limit on thickness (absolute value, or in % increase of the initial value). Default is 0.1%. Element reaching the limit will be removed from the sensitive set (thickness kept constant).
MINT=mint	Lower limit on thickness (absolute value, or in % decrease of the initial value). Default is 99.9%. Element reaching the limit will be removed from the sensitive set (thickness kept constant).
CONSTANT	Thickness of all elements in the SDOMAIN will be changed by a constant value (average of changes).

Examples:

```
sdomain(const)=pid(21 thru 30)
sdomain(max=1%, min=99%)=element(all)
sdom(max=5%)=set(bottom,side)
```

Remarks:

1. As many SDOMAIN commands as necessary are allowed.
2. SDOMAIN commands can be “mixed” i.e., some with and some without the CONSTANT parameter.
3. CONSTANT refers to constant change, not to constant thickness!
4. The ‘*export this iteration*’ feature transfers new thicknesses to the new NASTRAN input file as follows:
 - a) If CONSTANT parameter omitted the thickness is written to each CQUAD4 separately i.e., original PSHELL commands will be ignored by the analysis.
 - b) If CONSTANT parameter used on a SET defined by PID the thickness is changed in PSHELL cards.
 - c) If CONSTANT parameter is used on a SET defined by any other means then same as case a)

PROCESS(SIZING)

SPARAMETER

Modify sizing parameters

SPARAMETER(parameter) = value

Sets parameters specific to a PROCESS(SIZING) iterative design improvement.

Table 17 SPARAMETER Parameters and their default values

Parameter	Description	Default Value
<i>REMOVE</i>	Allows the removal of elements during the sizing process.	NO

Examples:

sparam(remove)=yes

Remarks:

1. SPARAM (remove) allows elements to be removed during the sizing process. It is not recommended when there is a stress response, but it can significantly speed up the analysis in most other cases.

PROCESS(TOPOLOGY)

PROCESS(TOPOLOGY) Block

Commands:

TDOMAIN *must consist of elements*
TPARAMETER

PROCESS(TOPOLOGY) is available for 2D and 3D models.

Only one PROCESS is permitted.

PROCESS(TOPOLOGY)

TDOMAIN

Specification of Model Regions to be utilised for topology

$$\text{TDOMAIN} = \left\{ \begin{array}{l} \text{SET(setnames)} \\ \text{PID(pids)} \\ \text{ETYPE(etypes)} \\ \text{ELEMENT(eids)} \end{array} \right\}$$

Defines the active part of the model to be utilized in topology.

Examples:

Tdomain =pid(21 thru 30)
tdomain=element(all)
tdom=set(bottom,side)

Remarks:

1. Only one TDOMAIN command is allowed.
2. The TDOMAIN must contain only either 2D or 3D elements.
3. The model may contain 1D, 2D and 3D elements in it.

PROCESS(TOPOLOGY)

TPARAMETER

Control Parameters for PROCESS(TOPOLOGY)

TPARAMETER(parameter) = value

Sets parameters specific to a PROCESS(TOPOLOGY) iterative design improvement.

Table 18 TPARAMETER Parameters and their default values

Parameter	Description	Default Value
<i>ACCELERATION</i>	Exponential term used on the current pseudo density to promote faster convergence. $6 \geq ACCELERATION \geq 1$	3
<i>CHECKERBOARD</i>	Checkerboard control on	No
<i>EXTERNALFIRST</i>	Solids with external surfaces, and shells with external edges, are removed at each iteration.	No
<i>REMOVE</i>	Specifies whether elements should be completely removed from the stiffness matrix when they reach <i>THRESHOLD</i>	Yes
<i>START</i>	The starting pseudo density applied to all elements $1 \geq START > THRESHOLD \geq 0.05$	0.6
<i>THRESHOLD</i>	Elements with pseudo density below this value are removed from the analysis. $0.5 \geq THRESHOLD \geq 0.05$	0.4
<i>TOOLDIRECTION</i>	x/y/z – a vector in global coordinates defining a tool/pull-out/cast direction manufacturing constraint.	n/a
<i>TOOLINVISIBLE</i>	Element set(s) that the tool direction will ignore.	n/a

Examples:

tparameter(threshold)=0.3

PROCESS(TOPOLOGY)

tparam(acc)=3

tparam(tooldir)=1.0/0.0/0.5

tparam(toolinvis)=set(engine_mount_lhs)

Remarks:

1. All elements that contain a constrained or loaded node are removed from the domain at the start of the process.
2. Elements that reach the pseudo density multiplier of 1 are automatically removed from the tdomain in order to speed up the process.
3. Setting DELETE=NO will result in a minimum pseudo density multiplier value of THRESHOLD.
4. Recommended value for CHECKERBOARD for predominantly tetra element models is YES.

APPENDICES

APPENDICES

ReSHAPE Structure Overview

Main Features

- NASTRAN Bulk Data Section is used without any modification.
- NASTRAN op2 binary result file is written by ReSHAPE for post-processing. (ReSHAPE can also output HyperMesh *.res, MEDINA *.bof, or ASCII output if required)

Note: ReSHAPE currently handles only one load case per analysis run.

Operating Sequence

ReSHAPE is intended for use with NASTRAN input decks. It is assumed that the input deck, before sensitivity analysis information is added, runs successfully in a NASTRAN SOL 101, 103 or 105 solution sequences. There is substantial error checking by ReSHAPE on the bulk data section of the deck, but it should not be regarded as comprehensive.

A list of supported NASTRAN Case Control Section and Bulk Data Section commands is given in Appendix A.

ReSHAPE does not use all the data in the fields of the NASTRAN Bulk Data entries. A warning message is issued if an unsupported value exists in a data field. The user may proceed with the analysis after the warnings; however they should be aware of the possible effects on the analysis results caused by omitting any features.

ReSHAPE processes the Case Control Section in the same way as NASTRAN, except that ReSHAPE only supports single SUBCASE analysis. ReSHAPE reads through the LOAD and SPC commands at the global level and then through the first SUBCASE section until sufficient definitions have been found. Similar to NASTRAN processing, the last appearing LOAD and SPC commands listed in the global or SUBCASE sections are retained as the active ones for the analysis. Also, when LOADs or SPCs are specified in the SUBCASE level these commands override any previous commands at the global level as the active SUBCASE commands.

If more than one SUBCASE block is active in the NASTRAN input file, the user has the option to select the required subcase by using the SUBCASE command in the CONTROL block of the command file.

APPENDIX 1

Structure of ReSHAPE Commands

ReSHAPE is invoked by reading a ReSHAPE command file corresponding to a NASTRAN input deck. ReSHAPE uses the command file of the same name as the NASTRAN input file specified if no command file name is specified with the **-cmf** runtime flag.

A command file consists of **commands**, each command starting at a new line. Each command is activated by its keyword.

Commands are organised in **blocks**. Each block is activated by a special command – a **block header**. With the exception of the first keyword – RESHAPE – and the last keyword – END, the blocks can be entered into the command file in any order.

With the exception of the RESHAPE and CONTROL blocks, each block must contain commands relevant to the function of the block only. Most of the block commands can be entered in any order, but have to remain inside the relevant block. The PROCESS block is an important exception; the order of its commands is of utmost importance.

The keywords of the command blocks – RESHAPE, CONTROL, RESPONSE, SIDECONDITIONS, PROCESS – represent distinct components that are pertinent to design improvement.

Descriptions of these blocks are given below with their associated keyword.

Reshape (RESHAPE) selects finite-element, sensitivity or improvement analysis. The type of improvement process is selected by the command PROCESS. The RESHAPE command must be followed by data commands GSET, ESET, and NSET, if these are required.

Control (CONTROL)

The CONTROL block controls global parameters i.e., those valid for the whole model such as mesh quality, iteration step control, global tolerances, etc.

Responses (RESPONSE)

1. Objectives

The objective is the response of the model the value of which is to be increased or reduced by a structural change. An objective may be, for example, displacement, stress, modal frequency or volume (mass, area, length). The direction of the response change is selected by specifying the *target* value. .

More responses can be defined as objectives; the order of their activation is defined by the value of the parameter STRATEGY on the RESPONSE command.

APPENDIX 1

2. Constraints

In a SENSITIVITY analysis only active constraints affect the solution i.e., those which are to be kept at their values in the initial model by parameter HOLD.

More than one command defining performance constraints is allowed (maximum 10).

Side Conditions (SIDECONDITIONS)

Side conditions are used to control the design shape changes and/or to set up coupled relationships between the design changes and various geometric constraints.

Model change and smoothing block (PROCESS)

The function of PROCESS block is to define the parts of the model for reshaping. This is done by the commands DOMAIN, PASSIVE, LOCKED and UNLOCKED. Only one PROCESS block is allowed in the current code. The PROCESS block has to contain one and only one DOMAIN command.

The domain is specified via predefined selectors: sets, property ID, element types, or as nodes. DOMAIN and PASSIVE commands can use only element type selectors. Commands LOCKED and UNLOCKED can use both element and node type sets.

The DOMAIN command must be the first command in the PROCESS block and may appear only once.

There may be more than one PASSIVE, LOCKED or UNLOCKED command, however, they have to be grouped together and the LOCKED group must follow the PASSIVE group (if present).

The additional commands in the PROCESS block, and a brief summary of each, is listed below.

Sensitivity PROCESS () or any PROCESS(type)

Sensitivities in nodes are calculated analytically by ReSHAPE. They are calculated as sensitivity densities by taking into account the element sizes. These sensitivities can be re-scaled into a nodal change. If PROCESS(type) is used, all other cards inside the block other than DOMAIN, PASSIVE and LOCKED commands are ignored. This gives the user a flexibility to use an already prepared PROCESS block, or to decide on the improvement method later, after the sensitivity calculations have been completed.

Raw PROCESS(RAW)

The difference between the SENSITIVITY and the RAW processes is that the latter is iterative, whereas the former gives only the sensitivity of the original model to a design objective. The RAW process is suitable for the iterative improvement of some structures such as truss type structures (frames).

APPENDIX 1

Cubic Geometry PROCESS (GEOMETRY)

The parametric coordinates of the elements are calculated by ReSHAPE, based on cubic splines fit to the selected sets of the existing model. The sets are defined in GSET commands in the CONTROL block.

The nodal displacements for the shape change are automatically smooth, on a ‘best fit’ basis with respect to the edges of the model.

Basis vector PROCESS (VECTOR)

The columns of the influence matrix are either generated internally by ReSHAPE, using displacements or modes of vibration of selected domains as the input, or they can be read from a file generated by the user.

Influence function PROCESS (FUNCTION)

The columns of the influence matrix are generated from three sources;

1. Influence functions “emanating” from user defined control points.
2. Internal influence functions, originating at the nodes of the model boundary to ensure zero nodal change.
3. User defined special constrains (e.g. FPLANE).

The columns generated from the last two sources are eliminated by satisfying the prescribed constrains.

Evolutions PROCESS (EVOLUTION)

The columns of the influence matrix are either generated internally by ReSHAPE, using displacements calculated by applying the sensitivity vector for the responses as direct NASTRAN FORCE loads.

Sizing PROCESS (SIZING)

Sizing calculates where there is unnecessary mass, and consequently reduces the thickness of the shell elements at this region, given specified constraints.

Topology PROCESS (TOPOLOGY)

APPENDIX 1

Topology changes pseudo-volume of the solid and shell elements of the DOMAIN. Elements with volumes below a prescribed threshold level can be automatically removed from the model.

Keyword and descriptor uniqueness

ReSHAPE requires that keywords and parameters be specified with sufficient characters to make a unique definition. For instance, **CONSTRAINT** is adequately specified by **CONS** since this is sufficient to distinguish it from the **CONTROL** keyword. Further examples are ELE to denote a collection of element IDs, ET to denote element types, ELS an element set, TA a target value, and P a property collector.

Case sensitivity

In general, ReSHAPE is not case sensitive. Therefore commands can be entered into the command file in lower case or upper case, or any combination of lower and upper cases. For example, the keyword **CONSTRAINT** is equivalent to **constraint** or **CoNsTraiNt**.

An exception is made in the case of user defined NSET and ESET set names - lower case and upper case names are interpreted as identifiers of distinct sets (ie. 'a1' and 'A1' could be two distinctly different sets). The definition and use of node and element sets is explained in Part 2 of this manual.

Table 19 summarises ReSHAPE supported characters for the definition of set names.

Alphabetic Characters	Numeric Characters	Special Characters
A to Z, a to z	0,1 to 9	+ _ @ # % &

Table 19 Characters supported in ReSHAPE set name definitions

Accuracy of input

The spelling of any sensitivity input, either in full or partial must be correct, for example **CIN**, **CONSTRIANT** will both generate errors. Character spaces in the sensitivity input are ignored by ReSHAPE, therefore the keyword **CONSTRAINT** can also be written as **C O N S T R A I N T** . Again the exception to this rule is the definition of ReSHAPE node and element set names which must be referred to as specified in an NSET or ESET command.

Command line format

The length of a line should not exceed 80 characters. If more than one line is needed to enter a data record, the user must include “,” (a comma) at the end of an incomplete command line to indicate that the data continues on the next line. Lines are terminated when trailing blank spaces follow a character other than a “,”. An example is given below.

```
DISPLACEMENT(TY,AVGW=95)=PIDS(5,6,7,8,9,  
10 THRU 55,77,78,79,100 THRU 1000)
```

APPENDIX 1

Within each command block, an objective response command, global parameters and collector commands can appear in any order and can be repeated as many times as appropriate. However, there may only be one command per line.

The tabulator (TAB key) can be used for indentations.

Comments

Comment lines can be inserted in any column by using a single “!” or “\$” before the comment characters. These are ignored by ReSHAPE in the same way as comment lines in NASTRAN.

The rules governing command file input format are summarised below:

- The command file must start with the keyword ReSHAPE.
- The keywords ESET, GSETS and NSET, if used, must immediately follow the keyword RESHAPE
- The command file is terminated with the END keyword.
- All blocks can appear in any order.
- The command file is not case sensitive, except for ReSHAPE set names.
- A keyword or descriptor must be defined with as many characters needed to make it unique.
- Character spaces and tabs are ignored.
- cA maximum of one command per line is permitted.
- Within each of the blocks command lines may appear in any order, with the exception of PROCESS blocks.
- Maximum allowable number of characters per line is 80, and per command 256.
- If a command spans multiple lines a continuation is denoted by a trailing ‘,’ comma on the incomplete line. The separation of the command line must occur at a place in the command that is coincident with the placement of a comma in the standard command format.
- Lines are terminated when trailing blank spaces follow a character other than a ‘,’ terminator.
- Comment lines can be inserted in the command file in any column or line. They are denoted by a preceding “!” or “\$” character
- Maximum length of an integer is 8 digits.

FORMAT OF ReSHAPE COMMANDS

Each command is described as follows:

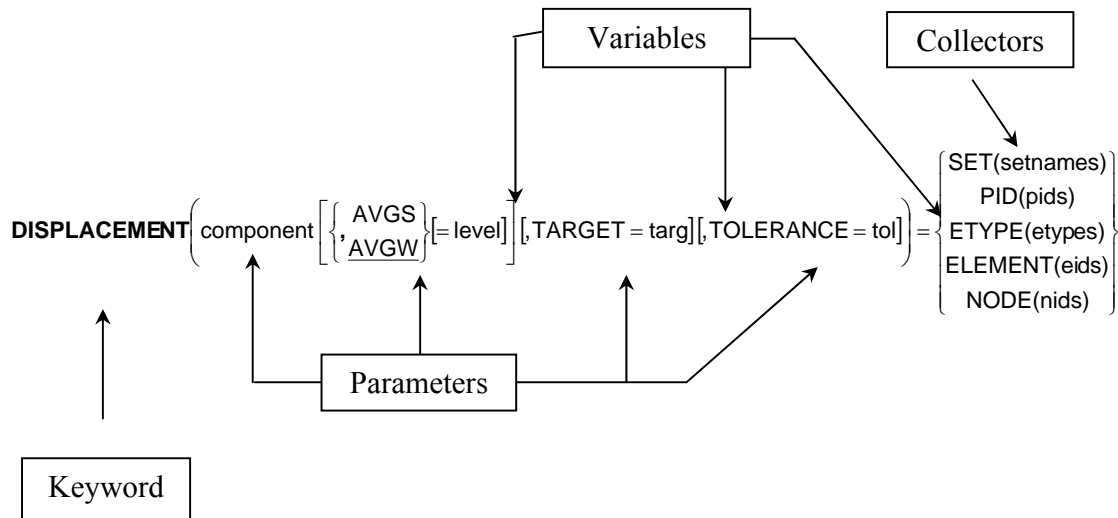


Figure 3 Format of a typical command and its component parts

Parameters printed in uppercase must appear as such in the input. Parameters in lower case represent variables for which the user must or may provide a value.

Braces { } indicate that a choice of parameters is mandatory. If the parameters are stacked vertically, then only one may be specified; if default is available, it is underscored.

Brackets [] indicate that a choice of Parameters is optional. If the Parameters are stacked vertically, then only one may be specified; if default is available, it is underscored.

Parameters called collectors are used to select the required entities using several methods.

Parameters and Meaning

Each of the Parameters is discussed briefly. The descriptor type (e.g. integer, real, or character), allowable range, and default value are enclosed in parentheses. The user must specify the descriptor, if no default is given.

Examples

Typical examples are included to demonstrate the correct usage of each keyword.

Remarks

The remarks are generally arranged in order of importance and indicate, for example, the relationship between commands, restrictions and recommendations on use of the command, any further information about the Parameters, and similar.

Running ReSHAPE

Input and output files

ReSHAPE is a batch program with NASTRAN files (*.bdf, *.nas, *.dat) used as input and NASTRAN .op2 files as default output. Both these files can be generated or processed by most graphics pre- and post-processors. For Medina and Hypermesh, ReSHAPE provides direct translators for output files (*.bof, *.res, resp.)

The commands to guide ReSHAPE are simple and easy to understand and use. They are written into a **command file** (text format). The use and format of commands are discussed in the following chapters of this document. A list of all ReSHAPE commands is given in Appendix 2. A detailed description of all ReSHAPE commands is given in the ReSHAPE Reference Manual.

Executing ReSHAPE in batch mode under UNIX/Linux

(assumes a symbolic link is created to the reshape script)

Basic commands

After the command file has been prepared and an input file containing the model definition has been created, ReSHAPE can be executed by typing:

```
reshape <filename>.<ext>
```

where <filename>.<ext> is the name of the input file specified by the user, <ext> being one of bdf, nas, dat or fem.

By default ReSHAPE will use the command file of the same <filename>, with <ext> being cmf i.e., <filename>.cmf, if it exists. A <filename> may consist of up to 70 characters.

To use the commands from a *command file having a different name from the model input filename* the following run-time syntax can be used:

```
reshape <filename>.<ext> -cmf <commandfilename>
```

In this case the results files produced will have the filename of the command file, i.e. <commandfilename> with a number of extensions. Command files can be edited independently of the model deck. This makes it easy to perform different analyses, sensitivity and improvement runs on the same model configuration.

APPENDIX 2

Output from ReSHAPE runs

The results are saved in a NASTRAN ***.op2** file (NASTRAN UNIX binary output) after every successful iteration step, by overwriting the previous results. This ensures that the user obtains results even if the iteration process is interrupted by some error, or by the user's interrupt from the keyboard.

To request an *output type different from the default*, ReSHAPE can be executed by typing:

```
reshape <filename>.<ext> -r <type>
```

Where the *type* parameter can have the following values: **op2** (NASTRAN, which is a default), **bof** (MEDINA), **res** (Hypermesh results file) and **plt9** (ASCII column wise format). Only the ***.op2** file will be saved after each successful iteration; the other result files will be saved only after the normal termination of the program. However, most post-processors have capabilities to translate **op2** files into their own formats.

New model and restarting

A new model based on your shape changes should be created from the user interface, rather than from the command line. It may be preferable to use your pre-/post-processor as well, if you want to apply very specific shape changes from your reshape results.

Suppressing the command file

It is possible that several **cmf** files may exist in the current working directory, but only an analysis run is required. The user may create a new command file containing instructions to perform analysis only or, more simply, use the following run-time option to *suppress the use of any command files thereby forcing an analysis run*.

```
reshape <filename>.<ext> -nocmf
```

Storing results under a name prescribed by the user

The *results may also be stored under a different name* from the both command file and model input file, using the following option:

```
reshape <filename>.<ext> -id <newfilename>
```

The following example reads commands from the file **<cmffilename>**, uses the input model **<filename>.<ext>** and stores the results as **<newfilename>.<op2>**.

```
reshape <filename>.<ext> -cmf <cmffilename> -id <newfilename>
```

User defined temporary directory

The *default directory* for the placement of temporary ReSHAPE files may be specified by

```
reshape <filename>.<ext> -tmp <dirname>
```

By default this directory is the directory from which ReSHAPE is executed.

APPENDIX 2

Listing of all run options

Complete information on ReSHAPE run options may be obtained by typing

reshape -info

This results in the following print:

ReSHAPE Information and Usage:

***reshape -r RESULT -tmp TMPDIR -cmf COMMAND -id IDENTIFIER
-nocmf -info input-file***

OR

reshape -u UTILITY

input-file

An input-file is required on a reshape command.
The type of input to be read is determined from the input-file extension. Input-file types are:
.bdf, .nas, .dat - nastran bulk data file.

RESULT

determines the type of result file to produce after the reshape analysis. Valid ***RESULT*** types are:
-r op2 - nastran results file(default).
-r bof - medina results file.
-r res - hypermesh results file.
-r plt9 - ascii files containing results.

TMPDIR

Will change the default directory used for temporary files from /tmp/ to TMPDIR/

COMMAND

By default the command file is assumed to be the same as the input-file with an extent of '.cmf'.
-cmf allows an alternate command file to be specified.

-nocmf

This parameter allows for an existing .cmf file to be ignored. That is, an ANALYSIS_ONLY run will be performed.

IDENTIFIER

By default all files created by ReSHAPE will be the same as the cmf-file with different extents.
The ***IDENTIFIER*** parameter will be used as the base name for creating all files.

APPENDIX 2

-info

parameter causes this information to be listed.

UTILITY

-u initiates a ReSHAPE utility program or process. When a process requires further information a series of questions will be prompted to the screen. The **-u** parameter also allows for additional(different) results files to be created without need to rerun long running analysis runs:

-u remove

- remove all but model input files.

-u combine

- create new **.osh** file from several **.osh** files.

-u extract

- list and extract available example files.

-u op2

- create nastran results file.

-u bof

- create medina results file.

-u res

- create hypermesh results file.

-u plt9

- create ascii files containing results.

Results file codes

The following result number codes appear in the output files (.op2, .res, .bof, .plt9)

CODE	TYPE	Original ReSHAPE file extension	Description
100	Geometry Shift	gsh	Change in shape resulting from spline fitting to patches when using PROCESS(GEOMETRY)
110	Displacement (Original Model)	ads	Nodal displacement results file from static analysis of original model
120	Stress (Original Model)	ast	Element stress results file from static analysis of original model
131,132, ...	Normal Modes (Original Model)	am1,am2,...	Mode shape result file from eigenvalue frequency analysis of original model (1 per mode)
161,162, ...	Buckling Modes (Original Model)	ab1,ab2,...	Mode shape result file from eigenvalue buckling analysis of original model (1 per mode)
200	Sensitivity (Original Model)	ssh	Nodal sensitivity results.
201	Sizing Results	ssz	Thickness sensitivity results
300	Shape Change	osh	Nodal movement results file from shape improvement, corresponds to new shape when added to original model
301	Sizing Results	osz	Improved thickness of shells
302	Sizing Results	osz	Thickness change of shells
310	Displacement	ods	Nodal displacement results file from static analysis of improved model
320	Stress (Improved Model)	ost	Element stress results file from static analysis of improved model
331,332, ...	Normal Modes (Improved Model)	om1,om2,...	Mode shape result file from eigenvalue frequency analysis of improved model (1 per mode)

ReSHAPE ↔ NASTRAN commands

In developing ReSHAPE, emphasis has been placed on supporting NASTRAN commands in ReSHAPE wherever possible. ReSHAPE is intended to perform sensitivity and improvement analysis with minimal specialised knowledge. A high proportion of the most commonly used NASTRAN cards for static and modal analyses, and the most commonly used fields in those cards, are supported in ReSHAPE. Special mention is made here of certain features in NASTRAN that are not supported in ReSHAPE and have the potential to alter the analysis model.

PARAM

ReSHAPE only supports the PARAM, INREL, -2 card for an analysis with inertia relief.

In some cases analyses some modifications to the NASTRAN deck may need to be carried out run the model in ReSHAPE. Particular attention should be paid to the use of the AUTOSPC command (the same will apply to the BAILOUT command) since this is commonly used in NASTRAN. AUTOSPC will add constraints to the model where nodes with inadequate stiffness or support are found. The use of this parameter may be required for a successful run on a particular model in NASTRAN. However ReSHAPE does not have any similar facility, and does not need it for the successful use of CQUAD4 and CTRIA3 elements. Therefore it is recommended that the user be aware of the implications of AUTOSPC in the NASTRAN model run before attempting a run in ReSHAPE.

Rigid Body elements

Rigid elements are supported. The NASTRAN RBE2 element is supported in all degrees of freedom on the independent node and either of the combinations 123456 or 123 on the dependent nodes. Chained combinations are supported, however ReSHAPE sorts these elements differently from NASTRAN, effectively eliminating nodes on elements between the independent node and dependent nodes. Therefore forces and masses should be referenced to and placed on the independent node in a chain combination. The order of definition of independent and dependent nodes is also important. ReSHAPE requires that the first node of an RBE2 definition be the independent node and the following nodes the dependent nodes. A dependent node can not have multiple independent nodes assigned to it. RBE3 is supported in a similar way.

RBE3 element must be loaded or be free. Displacements of the REFGRID node are not calculated.

APPENDIX 4

Supported NASTRAN commands

The following list contains NASTRAN Case Control Section and Bulk Data Section commands currently supported by ReSHAPE. For each command the fields supported are documented.

CASE CONTROL SECTION

Command	Comment
LOAD	Fully supported
SPC	Fully supported
METHOD	Fully supported
SUBCASE	Fully supported
TEMPERATURE	Fully supported

BULK DATA SECTION

Parameters

Command	Comment
PARAM,INREL,-2	ReSHAPE does not support -1, however your model is easily converted to -2 by removing all SPCs.

Nodes

Command	Comment
GRDSET	Supports fields 3, 7, 8
GRID	Supports fields 1 to 8 inclusive

Rigid Elements

Command	Comment
RBAR	Supports all fields
RBE2	Supports all fields
RBE3	Supports all fields, but not "UM" option

APPENDIX 4

Coordinate Systems

Command	Comment
CORD1C	Supports all fields (1 to 9 inclusive)
CORD1R	Supports all fields (1 to 9 inclusive)
CORD1S	Supports all fields (1 to 9 inclusive)
CORD2C	Supports all fields (1 to 14 inclusive)
CORD2R	Supports all fields (1 to 14 inclusive)
CORD2S	Supports all fields (1 to 14 inclusive)

1-D Elements

Command	Comment
BAROR	Supports fields 3 and 6 to 8 inclusive
BEAMOR	Supports fields 3 and 6 to 8 inclusive
CBAR	Supports fields 1 to 8 inclusive
CBEAM	Supports fields 1 to 8 inclusive
CBUSH	Supports fields 1 to 9 inclusive
CELAS1	Supports all fields (1 to 7 inclusive)
CELAS2	Supports fields 1 to 7 inclusive
CGAP	Supports fields 1 to 8 inclusive (not identical nodes)
CONM2	Supports fields 1 to 3,5,12,14,17
CONROD	Supports fields 1 to 7 inclusive
CROD	Supports all fields (1 to 5 inclusive)
PBAR	Supports fields 1 to 7 inclusive, 22 to 24 inclusive
PBEAM	Supports fields 1 to 8 inclusive, fields 2 and 3 (K1 and K2) on second-last continuation line
PBUSH	Supports fields 1 to 9 inclusive
PELAS	Supports fields 1, 2, 3, 6, 7
PGAP	Supports fields 2 and 5 only
PROD	Supports fields 1 to 5 inclusive

APPENDIX 4

Shell Elements (2D)

Command	Comment
CQUADR	Supports fields 1-7 and 14-17 inclusive
CTRIAR	Supports fields 1-6 and 14-16 inclusive
CQUAD4	Supports fields 1-7 and 14-17 inclusive
CTRIA3	Supports fields 1-6 and 14-16 inclusive
PSHELL	Supports fields 1 to 6 inclusive

Solid Elements (3D)

Command	Comment
CHEXA	Supports fields 1 to 12 inclusive
CPENTA	Supports fields 1 to 9 inclusive
CTETRA	Supports fields 1 to 7 inclusive
PSOLID	Supports fields 1 to 3 inclusive

NOTE: The number of supported fields consequently dictates that ReSHAPE does not support higher order shell and solid elements. Users must be aware or make adjustments to their models to correct for the reduced number of grid points.

Temperature Analysis

Command	Comment
TEMP	Supports all fields
TEMPD	Supports all fields

Modal Analysis

Command	Comment
EIGR	Supports fields 1, 2, 4, 5, 7
EIGRL	Supports fields 1 to 5 inclusive

Material Types

Command	Comment
MAT1	Supports fields 1 to 6 inclusive

APPENDIX 4

Loads

Command	Comment
FORCE	Supports all fields (1 to 8 inclusive)
FORCE1	Supports all fields (1 to 6 inclusive)
FORCE2	Supports all fields (1 to 8 inclusive)
GRAV	Supports all fields
LOAD	Supports all fields
MOMENT	Supports all fields (1 to 8 inclusive)
MOMENT1	Supports all fields (1 to 6 inclusive)
MOMENT2	Supports all fields (1 to 8 inclusive)
PLOAD	Supports all fields (1 to 7 inclusive)
PLOAD2	Supports all fields (1 to 9 inclusive)
PLOAD4	Supports fields 1 to 4 and 8 to 15 inclusive

Single Point Constraints, Enforced Displacements

Command	Comment
SPC	Supports all fields (1 to 8 inclusive)
SPC1	Supports all fields
SPCADD	Supports all fields
SPCD	Supports all fields (1 to 8 inclusive)

NASTRAN elements and ReSHAPE commands

The following NASTRAN elements are compatible with the following directive commands. Note: Elements marked with * cannot be post-processed in HyperMesh or Patran with ReSHAPE results. For these elements, use the PLT9 results format.

RESHAPE

ELEMENT TYPE	OBJECTIVE/CONSTRAINTS								DOMAIN	LOCKED
	DISPLACEMENT	STRESS ⁽¹⁾	REACTION	MASS	VOLUME	AREA	LENGTH	MODE ⁽²⁾		
CBAR *	✓	✓	✓	✓	✓		✓		✓	✓
CBEAM *	✓	✓	✓	✓	✓		✓		✓	✓
CBUSH	✓	✓	✓						✓	✓
CELAS1	✓	✓	✓				✓		✓	✓
CELAS2	✓	✓	✓				✓		✓	✓
CGAP	✓	✓	✓				✓		✓	✓
CONM2	✓		✓						✓	✓
CONROD	✓	✓	✓	✓	✓		✓		✓	✓
CROD	✓	✓	✓	✓	✓		✓		✓	✓
CQUAD4	✓	✓	✓	✓	✓	✓			✓	✓
CTRIA3	✓	✓	✓	✓	✓	✓			✓	✓
CQUADR	✓	✓	✓	✓	✓	✓			✓	✓
CTRIAR	✓	✓	✓	✓	✓	✓			✓	✓
CHEXA	✓	✓	✓	✓	✓				✓	✓
CPENTA	✓	✓	✓	✓	✓				✓	✓
CTETRA	✓	✓	✓	✓	✓				✓	✓
RBAR	✓		✓						✓	✓
RBE2	✓		✓						✓	✓
RBE3										

Index

<hr/>		ESET, element sets	4
2		ETYPE	
		SELECTION	26
		EXTERNAL, constraint	29
<hr/>		F	
		FCONTROL	
		PROCESS(FUNCTION)	57
		FPARAMETER	
		PROCESS(FUNCTION)	59, 94, 97
		FPLANE	
		PROCESS(FUNCTION)	61
<hr/>		G	
		GCORNERS	
		PROCESS(GEOMETRY)	68
		GLINE	
		PROCESS(GEOMETRY)	69
		GPARAMETER	
		PROCESS(GEOMETRY)	70, 100
		GRIB	
		PROCESS(GEOMETRY)	71
		GSET, geometry sets	5
<hr/>		I	
		IMPROVEMENT, ReSHAPE()	2
		INERTIA RELIEF	115, 116
<hr/>		L	
		LENGTH, response	30
		LOCKED	
		PROCESS(EVOLUTION)	88
		PROCESS(FUNCTION)	62
		PROCESS(GEOMETRY)	72
		PROCESS(RAW)	49
		PROCESS(VECTORS)	79
<hr/>		M	
		MASS, response	30
		MASTERSLAVE, sidecondition	41
		MODE, response	31
<hr/>		N	
		NASTRAN, control	13
		NLIMITS, sidecondition	42
<hr/>		A	
		ALIMITS, sidecondition	38
		ANALYSIS, control	10
		ANALYSIS, ReSHAPE()	2
		AREA, response	30
		AVERAGE, sidecondition	39
<hr/>		B	
		BUCKLE, response	31
		BUCKLING analysis	10
<hr/>		C	
		COMPLIANCE, response	23
		condensation	<i>See</i> SUPER
		CONTROL	7
		2DMODEL	9
		ANALYSIS	10
		CPARAMETER	12
		NSTRAN	13
		OOC	14
		REPLAY	15
		SOLVER	16
		STEPS	17
		SUBCASE	18
		SUPER	19
		VCREATE	83
		CONTROL, header	8
		CPARAMETER, control	12
<hr/>		D	
		DISPLACEMENT, response	24
		DOMAIN	
		PROCESS(EVOLUTION)	87
		PROCESS(FUNCTION)	55
		PROCESS(GEOMETRY)	67
		PROCESS(RAW)	48
		PROCESS(VECTORS)	78
<hr/>		E	
		ELEMENTS	
		SELECTION	26

APPENDIX 5

NODES	
SELECTION	26
NORMAL	
PROCESS(EVOLUTION)	89
PROCESS(FUNCTION)	63
PROCESS(GEOMETRY)	73
PROCESS(RAW)	50
PROCESS(VECTORS)	80
NSET, node sets	6

O

OOB, control	14
--------------	----

P

PASSIVE	
PROCESS(EVOLUTION)	90
PROCESS(FUNCTION)	64
PROCESS(GEOMETRY)	74
PROCESS(RAW)	51
PROCESS(VECTORS)	81
PID	
SELECTION	25
PROCESS Blocks	46
PROCESS(EVOLUTION)	86
DOMAIN	87
LOCKED	88
NORMAL	89
PASSIVE	90
UNLOCKED	92
PROCESS(FUNCTION)	54
DOMAIN	55
FCONTROL	57
FPARAMETER	59, 94, 97
FPLANE	61
LOCKED	62
NORMAL	63
PASSIVE	64
UNLOCKED	65
PROCESS(GEOMETRY)	66
DOMAIN	67
GCORNERS	68
GLINE	69
GPARAMETER	70, 100
GRIB	71
LOCKED	72
NORMAL	73
PASSIVE	74
UNLOCKED	76
PROCESS(RAW)	47
DOMAIN	48
LOCKED	49
NORMAL	50
PASSIVE	51
UNLOCKED	53
PROCESS(SIZING)	95, 98
SDOMAIN	96, 99
PROCESS(VECTORS)	77
DOMAIN	78
LOCKED	79

NORMAL	80
PASSIVE	81
UNLOCKED	82
VINPUT	84

R

REACTION, response	32
REPLAY, control	15
RESHAPE header	2
RESPONSES	
AREA	30
BUCKLE	31
COMPLIANCE	23
DISPLACEMENT	24
EXTERNAL	29
LENGTH	30
MASS	30
MODE	31
REACTION	32
STRESS	34
VOLUME	30
RESPONSES	20
RESPONSES, header	21
RLIMITS, sidecondition	43

S

SDOMAIN, PROCESS(SIZING)	96, 99
SELECTION, elements	26
SELECTION, etype	26
SELECTION, nodes	26
SELECTION, pid	25
SELECTION, set	25
SENSITIVITY, ReSHAPE()	2
SENSITIVITY, stype	25
SET	
SELECTION	25
SIDECONDITIONS	36
ALIMITS	38
AVERAGE	39
MASTERSLAVE	41
NLIMITS	42
RLIMITS	43
TAVERAGE	44
SIZING	95, 98
SOLVER, control	16
STEPS, control	17
STRESS, response	34
STYPE	
SENSITIVITY	25
SUBCASE, control	18
substructuring	See SUPER
SUPER, control	19

T

TAVERAGE, sidecondition	44
TDOMAIN	99
TPARAMETER	100

APPENDIX 5

U

UNLOCKED	
PROCESS(EVOLUTION)	92
PROCESS(FUNCTION)	65
PROCESS(GEOMETRY)	76
PROCESS(RAW)	53
PROCESS(VECTORS)	82

V

VCREATE, control	83
VINPUT, PROCESS(VECTORS)	84
VOLUME, response	30