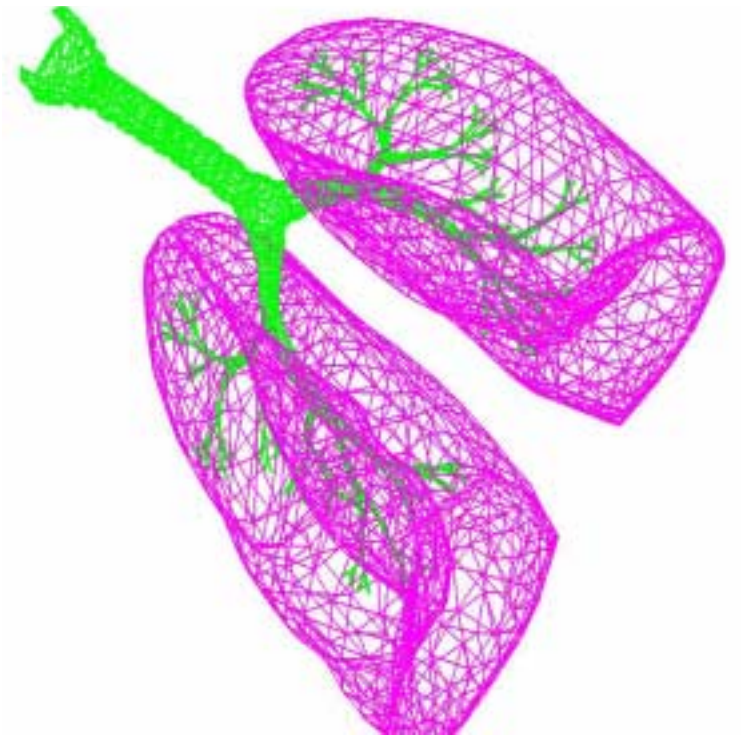
NWGrid
Command Reference

The *N*orth*W*est *Grid* Generation Code

Web Site: http://www.emsl.pnl.gov/nwgrid



Dr. Harold E. Trease
Lynn L. Trease
Applied Mathematics Group
Theory, Modeling, and Simulation Directorate
***William R. Wiley*, Environmental Molecular Sciences Laboratory**
Pacific Northwest National Laboratory
Operated by Battelle for the US Department of Energy

TABLE OF CONTENTS

addmesh 5
assign 7
box 8
bricktyp 8
check_amr_level 9
cmo 9
coggeshl 20
conenb 21
coordsys 22
copypts 22
cutreg 24
doping 24
dump 27
edit 30
eset 31
ether 33
extract 33
extrfl 37
field 38
filter 40
finish 40
flbound 41
geniee 42
gtg 42
help 43
hextotet 43
infile 44
interp 44
intersect 45
linkmaze 45
linksab 46
list 46
log 47
logadd 47
macro 48
macrodef 48
macroend 49
macrouse 50
mapgrid 50
mataddpt 51
matchbdy 51
matrec 51
memory 52
merge 52

modbld 53
modchk 54
model 54
modhelp 54
modpost 54
modprt 55
modset 55
modval 55
mregion 56
nb 58
newdict 58
newmesh 58
nn 58
nnblock 60
nnsurf 61
nntet 62
np 63
offsetsurf 63
parallel 64
partition 64
perturb 65
plotreg 66
pregion 67
process 68
pset 68
pstatus 72
pyoungs 73
quadxy 74
quadxyz 75
random 75
read 76
recon 77
refine 78
region 81
regnpts 83
resetpts 86
restart 87
rm 88
rmmat 89
rmpoint 89
rmregion 91
rmsphere 91
rmsurf 92
rotateln 92
rotatept 93
rz 94

rzbrick 95
rzd 97
rzdsab 98
rzs 99
rzst 100
sabangle 102
sabplot 102
sample 102
scale 103
scalec 104
search 105
setjtet 106
setpts 106
settets 108
smooth 109
source 112
sphbound 114
surf3d 115
tettest 116
title 117
trans 117
unit 118
variable 119
view3 120
wedge 121
window 121
write 122
x3d 122
z 123
zq 124
zqread 124
zqwrite 125
Documentation 126
Contact 126

addmesh

The **addmesh** command joins two meshes together at their common interface to produce a third mesh.

Formats

addmesh / amr / mesh3 / mesh1 / mesh2
addmesh / amr2 / mesh3 / mesh1 / mesh2
addmesh / append / mesh3 / mesh1 / mesh2
addmesh / delete / mesh3 / mesh1 / mesh2
addmesh / glue / mesh3 / mesh1 / mesh2
addmesh / merge / mesh3 / mesh1 / mesh2
addmesh / pyramid / mesh3 / mesh1 / mesh2
addmesh / add / mesh3 / mesh1 / mesh2 / [refine_factor] / **[tet | element | edges]**
addmesh / match / mesh3 / mesh1 / mesh2 / i1 i2 i3 i4 i5 i6
addmesh / match / mesh3 / mesh1 / mesh2 / rx1 ry1 rz1/ rx2 ry2 rz2 / rx3 ry3 rz3 / rx4 ry4 rz4 / rx5 ry5 rz5 / rx6 ry6 rz6
addmesh / intersect / mesh3 / mesh1 / mesh2 / pset_name

addmesh Parameters

Parameters	Type/Value	Description/Default
amr	**amr**	Use Adaptive mesh refinement to connect background mesh1 with submesh mesh2 and create mesh3.
amr2	**amr2**	
append	**append**	Append mesh2 to mesh1 and create mesh3. Similar to **merge** except imt, icr, itetclr of mesh2 have the value max(imt(mesh1)) added to mesh2.
delete	**delete**	Create mesh3 which is mesh1 with elements that intersect mesh2 deleted.
glue	**glue**	Synonym for merge.
merge	**merge**	Append mesh2 to mesh1 and create mesh3. Essentially this just concatenates two mesh objects.

addmesh Parameters

Parameters	Type/Value	Description/Default
pyramid	**pyramid**	Join a hex mesh to a tet mesh. The common surface must have matching nodes (i.e. there must be exactly two triangle faces on the tet grid that fit into one quad face of the hex grid). Pyramid elements will be constructed in the region where the two meshes join.
mesh3	char*32	Mesh object name for new mesh created. Becomes the new current mesh object. mesh3 = mesh1 + mesh2
mesh1	char*32	Mesh object name of the background mesh.
mesh2	char*32	Mesh object name of mesh to be added to mesh1.
add	**add**	Find the intersection of mesh1 and mesh2. Refine mesh1 where it overlaps mesh2 based on the refine_factor. After the refine steps have been done, the intersection of mesh1 and mesh2 is found, elements that overlap are deleted from mesh1 and mesh2 is appended to mesh1 to create mesh3.
refine_factor	real	refine_factor specifies the number of times that the background mesh will be refined. If this number is negative, or if it does not appear, then it will use the default. The default method determines the number of refinement iterations based on the volumes of the tets. It continues to refine until the elements on the interface boundary of the background mesh object are within a given factor (5.0) of the volume of the elements on the border of the incoming mesh object. This factor is a parameter constant called size_difference in the code continue_refinement.f. For example, if size_difference is set to 5.0, then the background mesh will be refined until the maximum volume element on the boundary with the incoming mesh object is no bigger than 5 times the volume of the maximum volume element on the border of the incoming mesh.
refine_type		The type of refinement that is executed.
	tet	Tetrahedral refinement is performed.
	element	Element based refinement is performed.

addmesh Parameters

Parameters	Type/Value	Description/Default
	edge	Edge based refinement is performed. This is the default.
match	**match**	Same as **merge** except the second mesh can be moved, rotated and translated. The first mesh does not move scale or rotate. If the interface needs to be scaled, translated and rotated that is accomplished by specifing 3 node numbers in each mesh or 3 node coordinates from each mesh that are to become coincident.
il i2 i3 i4 i5 i6	integers	Nodes are used.
rx1 ry1 rz1 … rx6 ry6 rz6	reals	Coordinates are used.
intersect	**intersect**	Create a **pset** called pset_name that contains all nodes in mesh 1 which intersect elements of mesh 2.
pset_name	char*32	Name of pset.

Special Notes

Care must be taken when using these commands because nothing is done to clean up the point type (itp) array after the **addmesh** operation. The user must often execute a series of **resetpts/itp** and **filter** commands to get the final desired result.

Some operations may only work with tet meshes**.**

Examples

addmesh / add / cmo3 / cmo1 / cmo2

assign

Assign a value to a code variable.

Format

assign / attrib_name / column / var_name / value

assign Parameters

Parameters	Type/Value	Description/Default
attrib_name	char*32	Name of mesh object attribute that are changing value of.

assign Parameters

Parameters	Type/Value	Description/Default
column		
var_name	char*32	Name of code variable that are changing value of.
value	real	New value.

Examples

assign / **-def-** / **-def-** / epsilonl / 1.e-4
assign / / / epsilonl / 1.e-9

box

Defines the six reflective boundaries that make up a box. The boundaries (planes) are defined in the following order xmin, xmax, zmin, zmax, ymin, ymax.

Format

box / xmin, ymin, zmin / xmax, ymax, zmax

box Parameters

Parameters	Type/Value	Description/Default
xmin, ymin, zmin	reals	Coordinates for the lower left corner.
xmax, ymax, zmax	reals	Coordinates for the upper right corner.

bricktyp

Defines the type of point distribution to be used in the next part of the brick mesh generation.

Format

bricktyp

Notes

The default is no brick mesh.

Currently, the only type of point distribution allowed is **rzst**. Currently, the only command that sets up a brick mesh is **regnpts**, and the region used in the **regnpts** command must include the boundaries of the region. The **bricktyp** variable must be set before **regnpts** is used if a brick mesh is desired. After all the **regnpts** commands have been processed, the commands, **filter, setpts, mass,** and **matchbdy**, must be placed in that order somewhere in the input file. The **search** command should not be used if a brick mesh is being set up.

check_amr_level

Format

check_amr_level ....

cmo

The **cmo** command operates on the Mesh Object (MO). There can be many Mesh Objects in the code for a given problem. Only one of these Mesh Objects may by the Current Mesh Object. There is also one Default Mesh Object which is used as the template for generating new Mesh Objects.

Formats

cmo / addatt / mo_name / att_name / type / rank / length / interpolate / persistence / ioflag / value

cmo / compress / mo_name /

cmo / copy / mo_name / master_mo /

cmo / copyatt / cmosink / cmo_src /attnam_sink / attnam_src

cmo / create / mo_name / [npoints / nelements / mesh-type]

cmo / create / mo_name / npoints / nelements / ndimensions_geom. / ndimensions_topo / nodes_per_element / faces_per_element / edges_per_element/

cmo / delatt / mo_name / att_name /

cmo / derive / mo_name / master_mo /

cmo / length / mo_name / att_name /

cmo / list

cmo / memory / mo_name / number_nodes / number_elements /

cmo / modatt / mo_name / att_name / field_name / new_field /

cmo / move / mo_name / master_mo /

cmo / newlen / mo_name /

cmo / printatt / mo_name / attribute_name / [print_opt] / [index_set]

cmo / readatt / mo_name / att_name / [att_name2] / operation / file_name
cmo / release / mo_name /
cmo / select / mo_name /
cmo / setatt / mo_name / att_name / index_set / value
cmo / status / mo_name /
cmo / verify / mo_name /

cmo Parameters

Parameters	Type/Value	Description/Default
addatt	addatt	Adds the Attribute, att_name, to Mesh Object, mo_name. See the modatt command for more details on the arguments. mo_name is required att_name is required
mo_name	char*32	Mesh Object name
att_name	char*32	Attribute name
type		Attribute type options:
	vdouble	Real*8 vector of rank rank with length vector elements (default).
	int	Integer vector of rank rank with length vector elements.
	vint	Single integer variable (length = 1, rank = 1 by definition).
rank		Attribute rank options:
	scalar	One entry per vector element (default).
	vector	Three entries per vector element.
	tensor	Nine entries per vector element.
	char*32	Any previously defined **int** attribute including user defined attributes may be used as rank.
length		Attribute length options:
	nnodes	(default)
	nelements	

cmo Parameters

Parameters	Type/Value	Description/Default
	char*32	Any previously defined **int** attribute including user defined attributes may be used as length.
interpolation		Attribute interpolation options:
	linear	Linear interpolation (default).
	constant	Constant value.
	copy	Copy values.
	sequence	Set to the node number.
	log	Logarithmic interpolation.
	asinh	Asinh interpolation.
	max	Set to the maximum.
	min	Set to the minimum.
	incmax	Set to the maximum plus one (**vint** attribute only).
	incmin	Set to the minimum plus one (**vint** attribute only).
	user	User provides.
	and	‘and’ the bits.
	or	‘or’ the bits.
persistence		Attribute persistence options:
	temporary	(default)
	permanent	
ioflag		**agx** (default) – options may be run together, with no space seperator.
	a	avs
	g	gmv
	f	fehm

cmo Parameters

Parameters	Type/Value	Description/Default
	s	sgi
	x	NWGrid
	no	
value	real	Default is 0.0
compress	compress	Shortens all memory managed arrays for Mesh Object mo_name to their actual lengths. Default for mo_name is the Current Mesh Object.
copy	copy	Makes an exact copy of Mesh Object, master_mo, including all data. The output Mesh Object, mo_name, will become the Current Mesh Object. If mo_name is the same as master_mo nothing happens. If mo_name exists it is over written. mo_name is required. Default for master_mo is the Current Mesh Object.
master_mo	char*32	Mesh object name.
copyatt	copyatt	Values are copied from cmosrc and attnam_src and written to cmosink and attnam_sink. There is currently no provision for indexed sets
cmosink	char*32	Current Mesh Object name to write values to.
cmo_src	char*32	Current Mesh Object name to copy values from.
attnam_sink	char*32	Attribute name to write values to.
attnam_src	char*32	Attribute name to copy values from.
create	create	Creates a new Mesh Object mo_name, which becomes the Current Mesh Object. If a Mesh is created using the mesh_type format, then values are supplied for the other. If mo_name exists nothing happens. mo_name is required.
npoints	integer	Number of points to allocate space for. Default is default mo.

cmo Parameters

Parameters	Type/Value	Description/Default
nelements	integer	Number of elements to allocate space for. Default is default mo.
mesh_type		If a Mesh is created using the mesh_type format, then values are supplied for the other parameters as follows:
		ndimndimnodes/faces/edges/ geomtopoelementelementelement
	tet	33446
	hex	338612
	pri(sm)	33659
	pyr(mid)	33558
	tri(angle)	32333
	qua(d)	32444
	hyb(rid)	33101012
	lin(e)	31221
ndimensions_ geom	integer	Default is default mo
ndimensions_ topo	integer	Default is default mo
nodes_per_el ement	integer	Default is default mo
faces_per_ele ment	integer	Default is default mo
edges_per_el ement	integer	Default is default mo

cmo Parameters

Parameters	Type/Value	Description/Default
delatt	delatt	Deletes the attribute att_name from Mesh Object, mo_name. Will not delete an attribute with a persistence of permanent. mo_name must be specified. att_name must be specified.
derive	derive	Uses Mesh Object, master_mo, as the template for deriving Mesh Object, mo_name. Mesh Object, mo_name, will be an image of master_mo but will contain no data. The output Mesh Object, mo_name, will become the Current Mesh Object. If mo_name is the same as master_mo nothing happens. If mo_name exists it is over written. mo_name is required. The default for master_mo is the Current Mesh Object.
length	length	Returns the memory length of attribute att_name for Mesh Object, mo_name. The default for mo_name is **-all-** The default for att_name is **-all-**
list	list	Returns the name of the Current Mesh Object and a list of all defined Mesh Objects.
memory	memory	Allows the user to preset the size of the memory managed arrays for Mesh Object, mo_name. mo_name is required. number_nodes is required. number_elements is required.
number_nodes	integer	Number of nodes.
number_elements	integer	Number of elements.
modatt	modatt	Modifies the field field_name for attribute att_name in Mesh Object mo_name. mo_name is required. att_name is required. field_name is required new_field is the type of the field, required Field_names (may be lower or upper case):

cmo Parameters

Parameters	Type/Value	Description/Default
field _name		Name of attribute field.
	name	Attribute name.
	type	Attribute type.
	rank	Attribute rank.
	length	Attribute length.
	interpolation	Attribute interpolation.
	persistence	Attribute persistence.
	ioflag	Attribute ioflag.
	default	Attribute default.
new_field	char*32	Field to be changed to in field_name.
move	move	Changes the name of Mesh Object, master_mo, to mo_name. The output Mesh Object, mo_name, will become the Current Mesh Object. If mo_name is the same as master_mo nothing happens. If mo_name exists it is over written. mo_name is required. master_mo default is the Current Mesh Object.
printatt	printatt	Print cmo attribute or category of attributes.
attribute_name	char*32	Name of valid cmo attribute.
	node	All attributes of nnodes length.
	element	All attributes of nelements length.
	-all-	All attributes of the cmo.
	-xyz-	xic, yic, and zic attributes of the cmo.
print_opt	value	Print attribute values (default).

cmo Parameters

Parameters	Type/Value	Description/Default
	list	Print attribute name along with its length.
	minmax	Print name, min and max of the attribute field, and its l length.
index_set	**pset**, get, pset_name	For attributes with length = nodes.
	eset, get, eset_name	For attributes with length = elements.
	ifirst, ilast, istride	For all other attributes, ifirst,ilast,istride must be integers; if blank then all members of the attribute will be modified.
readatt	readatt	Attribute to be read and added to or replacing existing nodes. Operation is either **add** which will cause new nodes to be added to the mesh object or ifirst, islast, istride which specifies the nodes whose values will be replaced. file_name is required. mo_name is required. att_nam is required.
att_nam2	char*32	Additional attribute names.
operation	add	New nodes to be added to the Mesh Object.
	ifirst, islast, istride	Specifies the nodes whose values will be replaced.
file_name	char*32	Specifies ASCII file to be read.
release	release	Deletes the Mesh Object mo_name. mo_name is required.
select	select	Selects Mesh Object mo_name to be the Current Mesh Object. If mo_name does not exist a new Mesh Object will be created using the Default Mesh Object as the template. mo_name is required.

cmo Parameters

Parameters	Type/Value	Description/Default
setatt	setatt	Sets the value of the specified attribute in the given range to the supplied value. This command requires that the mesh contains one or more nodes. mo_name is required. If att_name is blank, the active mesh object will be used.
value	integer or real	Value type depends on the type of the attribute.
status	status	Prints the status of Mesh Objects. mo_name default is **-all-**
verify	verify	Verifies that Mesh Object mo_name is consistent. mo_name default is **-all-**

Special Notes

CONVENTION: As a result of any command that generates a new Mesh Object, the newly generated Mesh Object becomes the Current Mesh Object.

RESERVED NAMES: The following names are reserved and may not be used for Mesh Object names:

-cmo- the Current Mesh Object
-default- the Default Mesh Object
-all- all Mesh Objects or Attributes
-notset- Null name for Mesh Object

Examples

cmo / addatt / cmo1 / boron1 / **vdouble** / **scalar / nnodes / asinh / permanent**
cmo / addatt / -cmo- / boron2 / **vdouble / scalar / nnodes / asinh / permanent** / gx / 2.0
cmo / addatt / cmo1 / boron3 / **vdouble / scalar / nnodes / user / temporary**
cmo / addatt / -default- / b
cmo / addatt / -default- / boron3

cmo / compress / mo_tet2
cmo / compress / -cmo-
cmo / compress
cmo / compress / -all-

cmo / copy / mo_tet2 / mo_tet1
cmo / copy / -cmo- / mo_tet1
cmo / copy / mo_tet2
cmo / copy / mo_tet2 / **-cmo-**

cmo / copyatt / cmosnk / cmosrc / **itetclr / itetclr**
cmo / copyatt / cmosnk / cmosrc **/ itetclr**
(both versions will copy itetclr field from cmosrc to cmosnk)
cmo / copyatt / cmosnk / cmosrc / **elevation**
cmo / copyatt /copyatt / cmosnk / cmosrc / elevation / zic /
(will copy the zic field of cmosrc to the elevation field of cmosnk)

cmo / create / mo_tet2
cmo / create / mo_tet2 / 0 / 0 / **hex**

cmo / delatt / mo_tet2 / boron
cmo / delatt / -cmo- / boron
cmo / delatt / -default- / boron
cmo / delatt / -cmo- / -all- (this will delete all attributes with persistence of temporary)

cmo / derive / mo_tet2 / mo_tet1
cmo / derive / -cmo- / mo_tet1
cmo / derive / mo_tet2
cmo / derive / mo_tet2 / **-cmo-**
cmo / derive / -default- / -cmo-
cmo / derive / mo_tet2 / **-default-**
cmo / derive /-default- / mo_tet1

cmo / length / mo_tet2 / boron
cmo / length / -cmo- / boron
cmo / length / -default- / boron
cmo / length / -cmo- / -all-
cmo / length / mo_tet2 / -all-
cmo / length
cmo / length / -all- / -all-
cmo / length / -all- / boron

cmo / list

cmo / memory / mo_tet2 / 1000 / 10000
cmo / memory / -cmo- / 1000 / 10000

cmo / modatt / mo_tet2 / boron / length
cmo / modatt / -cmo- / boron / length / **nnodes**
cmo / modatt / -cmo- / boron / **default** / 10.0
cmo / modat / -def- / boron / **default** / 10.0

cmo / move / mo_tet2 / mo_tet1
cmo / move / -cmo- / mo_tet1
cmo / move / mo_tet2
cmo / move / mo_tet2 / **-cmo-**

cmo / newlen / mo_tet2
cmo / newlen / -cmo-
cmo / newlen

print field values:
cmo / printatt / cmo1 / **zic** / 3, 8, 0
cmo / printatt / cmo1 / **itetclr** / **eset, get,** eset1

print attribute names:
cmo / printatt / cmo1 / **-all- / list**
cmo / printatt / cmo1 / **node / list**

print min and max of attribute fields:
cmo / printatt / cmo1 / **-all- / minmax**
cmo / printatt / cmo1 / **-xyz- / minmax**
cmo / printatt / cmo1 / **xic / minmax** / 7, 10, 0
cmo / printatt / cmo1 / **itp1 / minmax / pset**, get, psetr1

attribute **itp1** has length **nnodes** hence the **pset** syntax is valid
cmo / printatt / cmo1 / **itetclr / minmax / eset, get**, smallets

attribute **itetclr** has length **nelements** hence the **eset** syntax is valid

cmo / readatt / mo_tet2 / **xic, yic, zic** / add / / / myfile
new nodes will be added to mo_tet2, and their coordinates will be supplied from the file. The value of nnodes will be updated.
Contents of myfile:
0.017 12.65 7.25
1.1 10.2 3.4
will cause 2 nodes to be added to the mesh.

cmo / readatt / mo_tet2 **/ itp1** / newattr / **pset, set**, p1 / myfile

The values of itp1 will be replaced. If newattr does not exist it will be created as a vdouble node based attribute. If psetp1 contains the nodes 12, 27 and myfile has the contents:

```
0       100.2
        2       99.6
```

then itp1(12) will be set to 0, itp1(27) will be 2; newattr(12) will be set to 100.2 and newattr(27) will be set to 99.6.

cmo / release / mo_tet2
cmo / release / -cmo-

cmo / select / mo_tet2

cmo / setatt / cmo / **itp1** / 1, 0, 0 / 0
cmo / setatt / 3dmesh / **itetclr / eset, get**, blue / 3

Will set all elements in the element set 'blue' to have the value of itetclr to 3.

cmo / setatt / / ndimensions_geom / / / / 2

Will reset the ndimensions geometry attribute of the active mesh object to 2.

cmo / status / mo_tet2
cmo / status / -cmo-
cmo / status
cmo / status / -all-
cmo / status / -default-

cmo / verify / mo_tet2
cmo / verify / -cmo-
cmo / verify
cmo / verify / -all-
cmo / verify / -default-

coggeshl

Initialize one of the Coggeshall test problems.

Format

coggeshl / 2 / time0 / den0 / beta / gamma

coggeshl Parameters

Parameters	Type/Value	Description/Default
time0		
den0		
beta		
gamma		

conenb

Defines the reflective boundaries in (r,theta,phi)-geometry. First, the THETA sides are put on the spherical cone then so are the PHI sides. The THETA (in deg.) angle is in reference to the XY-plane and the PHI angle (in deg.) is taken with respect to the Z-axis.

Format

conenb / ibd / ibtp / radius / theta1, phi1 / theta2, phi2 / xcen, ycen, zcen

conenb Parameters

Parameters	Type/Value	Description/Default
ibd	integer	The reflective boundary number.
ibtp		Types of reflective boundary.
	2	Moving reflective boundary.
	3	Infinite reflective boundary (plane).
	4	Finite reflective boundary (forms a quad in 3D).
radius, theta1, phi1, theta2, phi2	reals	Limits of the cone.
xcen, ycen, zcen	reals	The vertex of the cone.

coordsys

This routine defines a local coordinate system to be in effect until another coordinate system is defined or the normal coordinate system is reset. The new coordinate system is defined by specifying an origin, a point on the new x-z plane and a point on the new z-axis. these points are specified in the normal coordinate system.

Format

coordsys / iopt / x0, y0, z0 /xx, xy, xz / zx, zy, z

coordsys Parameters

Parameters	Type/Value	Description/Default
iopt	define	Define a new local coordinate system mesh2.
	normal	Return to the normal coordinate system mesh2.
	save	Save the current coordinate system for recall mesh2.
	restore	Recall the last saved coordinate system mesh2.
x0, y0, z0	reals	The location of the new origin, mesh2.
xx, xy, xz	reals	A point on the new x-z plane and mesh2.
zx, zy, zz	reals	A point on the new z-axis. These points are defined with the normal coordinate system, and used only with the **define** option mesh2.

copypts

Copy a point distribution. There are two distinct forms of this command. The first format is designed to copy points from one mesh object into another. The second form duplicates points within a mesh object including all the attributes of the points.

Format

copypts / sink_cmo / source_cmo / 1st_sink_pt / sink_stride / 1st_source_pt / last_source_pt / source_stride / sink_attribute / source_attribute / **copy**

copypts / 1st_source_point / last_source_point / source_stride [/ 1st_sink_point / sink_stride / **copy**]

copyts / 1st_point / last_point / stride

copypts Parameters

Parameters	Type/Value	Description/Default
sink_cmo	char*32	If omitted, current mesh object will be used.
source_cmo	char*32	If omitted, current mesh object will be used.
1st_sink_pt, sink stride	reals	Used to restrict the copy to a subset of points. Points in the sink mesh object will be overwritten if sink_stride is not zero. If no sink points or sink stride are specified, then the copied points are placed at the end of the data arrays otherwise the copied points are written over the existing points starting at the 1st sink point.
1st_source pt, last source pt, source stride	reals	
sink_attribute, source attribute	char*32	Specify attribute fields. Attribute values will be copied from the source mesh object to the sink mesh object. For example the x-coordinate field in the source mesh object (xic) may be placed in the y-coordinate field of the sink mesh object. The user is warned that these values might not make sense in their new context.
copy	**copy**	
1st_point, last_point, stride	reals	

Note

The first form of the command gives the arguments sink first then source whereas the second form gives the source then the sink.

Examples

copypts / 3dmesh / 2dmesh /

Copy all points in 2dmesh to the end of the 3dmesh point list.

copypts / 3dmesh / 2dmesh / 0, 0/ **pset, get,** mypoints /

Copy the point set named mypoints from 2dmesh to the end of 3dmesh point list.

copypts / 3dmesh / 2dmesh / 100, 4 / pset, get, mypoints / boron / arsenic /

Copy the arsenic field from the point set named mypoints from 2dmesh replacing the boron field at every fourth point beginning at point 100 in 3dmesh**.**

copypts / pset, get, mypoints / 0, 0 /

Duplicate the point set named mypoints from the current mesh object and place the duplicated points at the end of the point list.

copypts / / / 0, 0 **/ pset, get, mypoints /**

Duplicate the point set named mypoints from the current mesh object and place the duplicated points at the end of the point list. Same effect as the example directly above. The current mesh object is used since the fields are blank on the command line

cutreg

Selects an existing region to be used as a cutting surface in the plotreg command.

Format

cutreg / region name

cutreg Parameters

Parameters	Type/Value	Description/Default
region_name	char*32	A blank or null region name turns off cutting surfaces.

doping

Creates a doping profile for the grid.

Format

doping / **constant** / field_out / **set** [**add** | **sub**] / 1st point, last point, stride / xcon

doping / **gaussian** / field_out / **set**[**add** | **sub**] /1st point, last point, stride / igeom / x1, y1, z1 / x2, y2, z2 / lateral_diffusion / concentration / standard_deviation /

doping / **table** / field_out / **set** [**add** | **sub**] / **linear** [**log**, **asinh**] / cmo_geometry / table_geometry /

doping / **constant** / field_out / **set**/

doping Parameters

Parameters	Type/Value	Description/Default
constant	constant	Profile is invariant over the specified region with value xcon.
field_out	char*32	Name of the field.
set add sub	set add sub	Indicates if the calculated or input-value added to, subtracted from or used to set the existing field node value.
1st point, last point, stride	reaks	Specify a point set restriction.
xcon		
gaussian	gaussian	Profile contains a bounding box (x1,y1,z1) to (x2,y2,z2) where the peak concentration will be. z1=z2 in 2D. The doping varies according to the gaussian distribution below.
igeom	integer	igeom = xyz [rtz, rtp]
x1, y1, z1 x2, y2, z2	reals	
lateral_diffusion		
concentration		
standard_deviation		
table	table	Reads in doping data from a DATEX2.1 format file.
linear log asinh	linear	
	log	
	asinh	

doping Parameters

Parameters	Type/Value	Description/Default
cmo_geometry		
table_geometry		
constant	constant	
set	set	

Notes

Gaussian distribution:

doping = concentration * exp(-(L/std_dev)**2)

where L is the effective distance and can be represented as:

L = sqrt(dy**2 + (1/lateral_diffusion)*(dx**2 + dz**2))

where

dy = y-y1 (or y2 for that matter)

dx = x-x1 if x < x1 < x2

= 0 if x1 < x < x2

= x-x2 if x1 < x2 < x

dz similar to dx.

If the values to be doped are integers, doping works in two ways. If the second field is integer1, the new nodal attributes are based on element material types. Use field_out and att_table_field set to imt1 in this case. The integer1 option is implemented only for setting node material (imt1). The imt1 values of the active mesh object nodes will be set by determining which element in the reference mesh object the node falls in. This element's material (itetclr) value is then assigned to the node imt1.

The integer2 option sets node based attributes in the active mesh object by determining which voronoi cell in the reference mesh object the node falls in. Then the value for the node corresponding to this voronoi cell is copied to the active node.

If the second field is integer2, the new nodal attributes are based on the table attribute types using the Voronoi cells around the table nodes.

For integer doping, function can be min or max to choose what happens if a cmo_out node falls on a boundary between two elements or Voronoi cells. For 3d, Voronoi cell based doping, function can also be minp or maxp which makes any cmo_out nodes that fall outside the cmo_table geometry set to the maxium number of materials plus one. Mapset can be set to create, use, or left blank. If create is used then an idop attribute is formed that maps the cmo_table nodes to the cmo_out nodes. If use is used, doping will read and use this previously formed and saved mapping. Note that doping of integers should be done without child/parent relationships. If parents

exist, the doping results are unpredictable at interface boundaries because the value of parent nodes are unpredictable there.

Examples

doping / **constant** / ric / set / pset, get, doping / gaussian / ric / add / pset
doping / table / pic / set /1, 0,
doping / table / Saturation / set / 1, 0, 0 / cmo_course / linear / zx / yx /
doping / integer1 / imt1 / set / 1, 0, 0 / cmo_old / imt1 / min
doping / integer2 / rad2 / set / 1, 0, 0 / cmo_old / rad1 / min / create

dump

This command produces an output file from a Mesh Object.

Formats

dump / file_type / file_name / [cmo_name]
dump / **x3d** | **gmv** / file_name / [cmo_name] / [**ascii** | **binary**]
dump / **avs** / file_name / [cmo_name] / iopt_points / iopt_elements / iopt_values
dump / geom | **recolor /** file_name
dump / coord | **stl** | **chad** | **datex** | **stl** / [cmo_name]
dump / zone / material_zone | outside_zone
dump / fehm / file_name / [cmo_name] / file_output / / coeff_output / [**delatt** | **ascii** | **keepatt**]

dump Parameters

Parameters	Type/Value	Description/Default
file_type	x3d	Writes a restart file that contains geometry and mesh object information. cmo_name can be **-all-** in which case all mesh objects are written to the file or it can specify a list of mesh objects to be written. The default is **ascii**. A subsequent **read / x3d** command will restart the code at the state at which the dump command was issued.
	gmv	Write a file to be read by the graphics program GMV. The defaults are binary and current mesh object.
	avs	Output in AVS UCD (Unstructured Cell Data) format. One can turn on or off the output of node coordinates (iopt_points), element connectivity (iopt_elements) and node attributes (iopt_values). 1 (default) is on, 0 is off.

dump Parameters

Parameters	Type/Value	Description/Default
	coord	Will output a file with node list x, y, z values and element connectivity list.
	zone	Will output a set of 3 files. file_name_**material**.**zone** is node list for each material (imt) value. file_name_**outside**.**zone** is a node list for each of 6 (top , bottom, left, right, front, back) possible external boundaries for rectangular geometries. Note that a node can belong to more than 1 list. For example a corner node can belong to both the front and the side.
	chad	
	fehm	Will call the **dump / coord** and **dump /zone** options first. The main function is to form a sparse coefficient matrix used for solving PDE on a triangular or tetrahedral delaunay grid.
	stl	Output in STL, stereo lithography format. This is only valid for triangular sheets.
	geom	Will write an ascii file containing the geometry information for the current run. This information includes the region and mregion definitions and surface, names, types and definitions.
	recolor	This command writes the existing **colormap** to the specified file. (See **colormap** command)
file_name	char*32	
	-all-	
	material_zone	Parameter only used with file_type **zone**. See file_type **zone** above.
	outside_zone	Parameter only used with file_type **zone**. See file_type **zone** above.
cmo_name		

dump Parameters

Parameters	Type/Value	Description/Default
ascii | binary	ascii binary	Parameter only used with file_types **x3d**, **gmv, avs.**
file_output		Parameter only used with **fehm** file_type.
	binary	Output sparse matrix in Fortran unformatted format. Default
	ascii	Output sparse matrix as ASCII format.
	binaryc	Output sparse matrix as Fortran unformatted format with a compression algorithm applied to the area coefficients.
	asciic	Output sparse matrix as ASCII format with a compression algorithm applied to the area coefficients.
coeff_output		Parameter only used with **fehm** file_type.
	scalar	Area/distance coefficients are output as scalars. Default
	vector	Area/distance. Coefficients are output as vectors
	both	Area/distance. Coefficients are output as scalars and vectors.
	area_scalar	Area. Coefficients are output as scalars.
	area_vector	Area. Coefficients are output as vectors
	area_both	Area. Coefficients are output as scalars and vectors.
	alternate_scalar	Area/distance. Coefficients are output as scalars with compression of coefficients and applied during calculation of coefficients.
delatt | ascii | keepatt		Parameter only used with **fehm** file_type.
	delatt	No new node attributes are created. Default.
	ascii	Output sparse matrix as ASCII format

dump Parameters

Parameters	Type/Value	Description/Default
	keepatt	Six node attributes are created (top, bottom, left_w, right_e,back_n, front_s) which are assigned values according to the direction of the octant of their normal vector.

Example

dump / **gmv** / gmv.out / 3dmesh
dump / x3d / x3d.out / **-all-** / **binary**

edit

Prints an edit of various quantities based on the value of the option argument, the point limits, and/or a material specification.

Formats

edit / iopt / ifirst, ilast, istride / material_#_or_name/
edit / **angular** / ifirst, ilast, istride / material_#_or_name / xcen, ycen, zcen
edit / **radial** / ifirst, ilast, istride / material_#_or_name / xcen, ycen, zcen
edit / points / ifirst, ilast, istride / material_#_or_name / array1, array2, array3, array4

edit Parameters

Parameters	Type/Value	Description/Default
iopt		Specifies what to print.
	No value	Edit of sums, averages, and extrema of position coordinates (x,y,z), velocity components (u, v, w), pressure (p), density (r), temperature (t), magnitude of velocity (umag), and of mesh object attribute fields.
	two	Gives same information as the default, but only for the two points specified.
	parts	Gives a list of materials types, their names, count and sequence.

edit Parameters

Parameters	Type/Value	Description/Default
	partsmry	Edit of sums by material for mass (TM), volume (TV), momentum (TX,TY,TZ), kinetic energy (TK), internal energy (TI), radiation energy (TR), total energy (TE), centers of compression (CX, CY, CZ) and the integral RHO*DR(RD).
	points	Lists up to 4 cell-center array values for a set of points. Possible array values are: xic, yic, zic, uic, wic, vic, pic, ric, eic, tic, tric, pric, xopic, etotic, xmic, partic, volic, csqartic, csqic, dellic, dedmric, sxxic, syxic, szxic, syyic, szyic or mesh object attribute name.
	angular	Total angular momentum about xcen, ycen, zcen.
	radial	Radial position (R) and velocity (QR) about xcen, ycen, zcen.
ifirst, ilast, istride	char*32	
material_#_or_name	char*32	
xcen, ycen, zcen	reals	
array1, array2, array3, array4	reals	

Examples

edit / **parts** /
edit /
edit / **points** / **pset**, **get**, some+points

eset

This command creates element sets (esets) with membership criteria based on relational operators or other criteria.

Format

eset / eset_name / element_attribute_name / relation_oper **/** value
eset / eset_name **/** logic_oper **/** eset_list
eset / eset_name **/ inclusive** | **exclusive / pset get** pset_name
eset / eset_name **/ region** | **mregion /** region_name | mregion_name
eset / eset_name **/ volume /** relation_oper / value
eset / eset_name

eset Parameters

Parameters	Type/Value	Description/Default
eset_name	char*32	Name of new eset.
element_attribute_name	char*32	Attribute name comparing value to.
relation_oper	eq ne lt gt **le** ge	Creates element sets equal to, greater than or less than value of an element attribute.
value	real	Value comparing against.
logic_oper	union inter not delete	Creates element sets of the union, intersection, not, or delete of esets.
eset_list	char*32	
inclusive | exclusive	inclusive | exclusive	Inclusive creates element sets of all elements <u>any</u> of whose nodes are in pset. Exclusive creates element sets of all elements <u>all</u> of whose nodes are in pset.
pset get	pset get	See **pset** command.
pset_name	char*32	Name assigned to existing pset.

eset Parameters

Parameters	Type/Value	Description/Default
region | mregion	region | mregion	Creates element sets by calculating the center of mass of the element and determines which mregion or region the center lies in. It is possible if the interface surfaces are curved that the center will not lie in the same mregion or region as the vertices. Using itetclr will give the better result.
region_name | mregion_name	char*32	Name assigned to existing region or mregion.
volume	volume	Creates element sets based on quality criteria (volume or aspect ratio).

Examples

eset / element_set1 / **itetclr** / **eq** / 4
eset / element_set2 / **inclusive** / **pset** / **get** / mypset
eset / element_set3 / **region** / upper
eset / element_set4 **/ volume / lt /** 3.0

ether

Distribute boundary velocity in region or volume

Format

ether / [region_name]

ether Parameters

Parameters	Type/Value	Description/Default
region_name	char*32	Name of region.

extract

This command produces a 2D mesh object from a 3D mesh object. A material interface, a plane , an iso-surface, a network, or an external surface may be extracted.

Formats

extract / **plane** / **threepts** / x1, y1, z1 / x2, y2, z2 / x3, y3, z3/ ifirst, ilast, istride / cmoout / cmoin

extract / **plane** / **ptnorm** / x1, y1, z1 / xn, yn, zn / ifirst, ilast, istride / cmoout / cmoin

extract / **plane** / **axes** / xv, yv, zv / ifirst, ilast, istride / cmoout

extract / **plane** / **abcd** / a,b,c,d / ifirst, ilast, istride / cmoout / cmoin

extract / **isosurf** / var / value / ifirst, ilast, istride / cmoout / cmoin

extract / **intrface** / region1 / ifirst, ilast, istride / cmoout

extract / **intrfac2** / region1 / region2 / ifirst, ilast, istride

extract / network / ifirst, ilast, istride / cmoout / cmoin

extract / external / region1 / ifirst, ilast, istride / cmoout / cmoin

extract Parameters

Parameters	Type/Value	Description/Default
2nd argument	plane	May be defined by three points in the plane, by a vector normal to the plane, by three points on the axes of the space, or by the coefficients of the plane equation ax+by+cz=d. In the case of a plane extracted along all or a portion of a material interface, only those points that lie inside the material (i.e.: away from the direction of the normal) will be picked up.If the extraction is on a boundary, the normal to the extraction plane must point out of the material in order for points to be picked up.
	isosurf	Is defined by the value of the surface and the mesh object field to test for this value. The output MO will be oriented such that the normals for the triangles point in the direction of increasing field.
	intrface	Is defined by the material(s) bounding the interface. The output MO triangles will be oriented the same as the triangles extracted from region1 of the input MO.
	intrfac2	

extract Parameters

Parameters	Type/Value	Description/Default
	network	Is defined as the material interface network from the mesh object. Unlike the other extract forms, this extracts the "parent" interface and not the "child" interface, and ignores boundaries. The output cmo also contains a new attribute "map" which maps nodes in the output cmo to (parent) nodes in the input cmo. This is supposed to work for 2D meshes (tri, quad, or hybrid) in 2D or 3D and 3D meshes (tet, pyr, pri, hex or hybrid).
	external	
3rd argument	threepts	The points must not be colinear.
	ptnorm	
	axes	
	abcd	
x1, x2, x3, y1, y2, y3, z1, z2, z3		Define points in the plane.
ifirst, ilast, istride	reals	First point, last point, and stride.
	pset, get, setname	The **pset** syntax can be used to limit the action of the extract with the interface and network options only. The other options ignore **pset**.

extract Parameters

Parameters	Type/Value	Description/Default
cmoout	char*32	The output 2D mesh object. The output MO will be oriented such that the outward normal of the plane that defines the surface will point in the same direction as the normals for the triangles in the output MO. If the command extracts on an isosurface, the output MO will be oriented such that the normals for the triangles point in the direction of increasing field. If the command extracts on an interface, the output MO triangles will be oriented the same as the triangles extracted from region1 of the input MO. In the case of a plane extracted along all or a portion of a material interface, only those points that lie inside the material (i.e. away from the direction of the normal) will be picked up. If the extraction is on a boundary, the normal to the extraction plane must point out of the material in order for points to be picked up. This will be a triangular grid. Node quantities will be renumbered so that the output MO will have nodes numbered consecutively starting at 1. It will also have triangles numbered consecutively from 1.
cmoin	char*32	The input 3D mesh object or the existing MO at the time the command is called. To select a specific MO, that is not the current one, precede the extract command by a **cmo / select** command. This must be a tetrahedral grid.
x1, y1, z1, xn, yn, zn	reals	Are components of a vector normal to the plane.
xv, yv, zv	reals	Are the values at which the plane crosses the corresponding axis.
a, b, c, d	reals	Are the coefficients in the equation of a plane: ax + by_+ cz + d = 0
var	char*32	The name of the field variable on which to make the isosurface.
	pic	
	ric	

extract Parameters

Parameters	Type/Value	Description/Default
	eic	
	uic	
	vic	
	wic	
value	real	
region1	char*32 region name or mregion name	The material numbers or the material region names whose interface is to be extracted.
	-all-	To extract from all interfaces.
region2	char*32 region name or mregion name	The material number or material region name for the second material. In this case, only those parts of the incoming MO interface that are on the surface of region1 and lie across from region2 will be extracted.
	-all-	To extrace from all. interfaces.

extrfl

Extracts a list of points contained in a region and changes the point types to be reflective boundary points (ipt=10).

Format

extrfl / ifirst, ilast, istride / xmn, ymn, zmn / xmx, ymx, zmx / icrnum

extrfl Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride	integers	The point-sequence numbers to search.

extrfl Parameters

Parameters	Type/Value	Description/Default
xmn, ymn, zmn xmx, ymx, xmx	reals	Define the reigion from which points will be extracted.
icrnum	integer	Constraint number to be set to points.

field

The **field** command option manipulates one or more specified fields in the Current Mesh Object.

Format

field / **compose** / composition function / ifirst, ilast, istride / field list
field / **mfedraw** / root file name / x1, y1, z1 / x2, y2, z2 / field list
field / scale / scale option / factor / ifirst, ilast, istride / field list
field / **volavg** / averaging option / iterations / ifirst, ilast, istride/ field list

field Parameters

Parameters	Type/Value	Description/Default
2nd argument	compose	For all points in the specified point set, we compose the field value with the specified composition function.
	mfedraw	Causes a binary dump of the specified fields to two files in the **mfedraw** input format. **mfedraw** is a graphics package for visualizing moving piecewise linear functions of two variables, such as those originally encountered in Moving Finite Elements. The files are named 'root1.bin' and 'root2.bin', where 'root' is the root file name argument. Because the graphics data are a function of two variables, you must supply two orthonormal vectors (x1,y1,z1) and (x2,y2,z2) which specify the graphics coordinate axes.
	scale	Scales the field values of the specified points.

field Parameters

Parameters	Type/Value	Description/Default
	volavg	For all the members of the point set and for all specified fields, replaces the point field values with values that represent the average of the field(s) over the control volumes associated with the points.
composition_ function		Example, if 'i' is in the point set and **asinh** is the composition function, we have the assignment: field(i) = asinh(field(i)).
	asinh	
	log	
ifirst, ilast, istride	integers	
field_list		
root_file_na me	char*32	
x1, y1, z1, x2, y2, z2		Two orthonormal vectors (x1,y1,z1) and (x2,y2,z2) which specify the graphics coordinate axes. More precisely, given 3D coordinates (x,y,z), the 2D graphic coordinates will then be (x*x1+y*y1+z*z1 , x*x2+y*y2+z*z2). For example, the choice: /x1,y1,z1/x2,y2,z2/ = /1.,0.,0./0.,1.,0./ causes the 'z' coordinate to be discarded while the 'x' and 'y' coordinates are unchanged.
scale_option	normalize	Multiply all the field values by **factor**. This has the effect of normalizing the field so that the new difference between the maximum and minimum values is equal to **factor**.
	multiply	Multiply all the field values in the point set by **factor**.
	divide	Divide all the field values in the point set by **factor.**
factor	real	(fieldmax-fieldmin), where 'fieldmax' and 'fieldmin' are the maximum and minimum values taken over the point set.

field Parameters

Parameters	Type/Value	Description/Default
averaging_option		The averaging option specifies what kind of control volume is to be used. The affect of this process is to broaden and smooth the features of the field(s), similar to the effect of a diffusion process.
	voronoi	Produces a diffusive effect independent of mesh connectivity. It requires that the mesh be Delaunay, or incorrect results will occur. (Not true for median.)
	median	
iterations	integer	Specifies a repeat count for how many times this procedure is to be performed on the field(s).

Examples

field / **compose** / **asinh** / 1, 0, 0 / **field** / **scale** / **normalize** / 4.

filter

Used to filter (delete) points that are too close, closer than the tolerance specified by the user, or duplicate points. This command records the deleted points as dudded out points (**itp**=21) but does not remove them from the point list.

Format

filter / ifirst, ilast, istride / [tolerance]

filter Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride	integer	Point sequence number. At least one point must be specified in order for this command to work properly.
tolerance	real	Default <= 1.e-16. Specifies how close the points can be.

finish

Terminate processing this command stream and return to the driver routine.

Format

finish

flbound

Files the quantities associated with a reflective boundary.

Format

flbound / 1 / ibd / ibtp / x1,y1,z1/x2,y2,z2/x3,y3,z3
flbound / 2 / ibd / ibtp / u1,v1,w1/u2,v2,w2/u3,v3,w3
flbound / 3 / ibd / ibtp / press, rho, einter, temp

flbound Parameters

Parameters	Type/Value	Description/Default
iopt	1	File the reflective boundary type (IBTP) and the three coordinates [(x1,y1,z1) - (x3,y3,z3)] for the IBD reflective boundary.
	2	File the reflective boundary type (IBTP) and the three velocity triplets [(u1,v1,w1) - (u3,v3,w3)] for the three corners of the IBD reflective boundary.
	3	File the reflective boundary type (IBTP) and the pressure,density,internal energy, and temperature for the IBD reflective boundary.
ibd		The reflective boundary number.
ibtp	2	Moving reflective boundary.
	3	Infinite reflective boundary (plane).
	4	Finite reflective boundary (forms a quad in 3D).
x1, y1, z1, x2, y2, z2, x3, y3, z3 u1, v1, w1, u2, v2, w2, u3, v3, w3	reals	Coordinates and velocities are specified at the three points of a plane.

flbound Parameters

Parameters	Type/Value	Description/Default
press, rho, einter, temp	reals	Physical properties are applied to the whole plane.

geniee

Generate element connectivity list (**jtet**) that gives neighbor information. Element connectivity is maintained by ***NWGrid***, but can also be generated by the user with this command.

Format

geniee

gtg

Grid-to-grid mapping function.

Format

gtg / **hybrid** / cmo_name
gtg / **amrtox3d** / cmo_name

gtg Parameters

Parameters	Type/Value	Description/Default
hybrid	**hybrid**	
cmo_name	char*32	
amrtox3d	**amrtox3d**	

help

Access help package.

Format

help / [variable_name | command_name]

help Parameters

Parameters	Type/Value	Description/Default
variable_name	char*32	Returns the variable definition and the correct value of the variable.
command_name	char*32	Returns the command description.

Examples

help
help / surface
help / **ipointi**

hextotet

Create a tetrahedral grid from a hexahedral grid or a triangle grid from a quadrilateral grid

Format

hextotet / ioption / cmo_tet / cmo_hex

hextotet Parameters

Parameters	Type/Value	Description/Default
ioption		Determines how the conversion is performed.
	2	2 triangles per quad, no new points.
	3	3 tets per prism, no new points.
	4	4 triangles per quad, 1 new point per quad.
	5	5 tets per hex, no new points.
	6	6 tets per hex, no new points.

hextotet Parameters

Parameters	Type/Value	Description/Default
	18	18 tets per prism, 6 new points(1 + 5 faces)
	24	24 tets per hex, 7 new points(1 + 6 faces).
cmo_tet	char*32	Name for new tet CMO.
cmo_hex	char*32	Name of existing hex CMO.

Examples

hextotet / 24 / cmo_tet / cmo_hex

infile

These commands instruct ***NWGrid*** to begin processing commands from a file. The **infile** commands may be nested. Each set of commands should be terminated with a **finish** command.

Format

infile / file_name

infile Parameters

Parameters	Type/Value	Description/Default
file_name	char*32	Name of ***NWGrid*** input file to be run.

interp

Used to interpolate to a given (logical) line, plane, or volume.

Format

interp / dimension_option / xmin,ymin,zmin / xmax,ymax,zmax

interp Parameters

Parameters	Type/Value	Description/Default
dimension_option		
xmin, ymin, zmin xmax, ymax, zmax	reals	

intersect

Creates a new Mesh Object from the intersection of two existing Mesh Objects. The existing Mesh Objects have to be topologically 2D and geometrically 3D. The created Mesh Object will be topologically 1D and geometrically 3D. Node quantities for the new Mesh Object will be create by interpolation on the corresponding node quantities of the first input Mesh Object, cmo_1_in.

Format

intersect / cmo_out / cmo_1_in

intersect Parameters

Parameters	Type/Value	Description/Default
cmo_out	char*32	
cmo_1_in	char*32	

linkmaze

Provides a link to the MAZE generator of DYNA.

Format

linkmaze / file name

linkmaze Parameters

Parameters	Type/Value	Description/Default
file_name	char*32	Default is mazelink.

linksab

Provides a link to the SABRINA.

Format

linksab / file name

linksab Parameters

Parameters	Type/Value	Description/Default
file_name	char*32	The default is sabdmp.

list

Prints a list of selected problem information.

Formats

list / list_info
list / pset / name

list Parameters

Parameters	Type/Value	Description/Default
list_info	variable	List mass point array variables and descriptions. This is the default.
	material	List material no., name and no. of ctive points per material.

list Parameters

Parameters	Type/Value	Description/Default
	pset	List pset set names.
name	char*32	List named pset point list.

log

Turn the batch output file and tty output file off and on.

Format

log / bat | tty / on | off

log Parameters

Parameters	Type/Value	Description/Default
bat | tty	**bat**	The output file called outx3dgen.
	tty	Prints to and reads from the user's screen.
on | off	**on**	Default is on for both **bat** and **tty**.
	off	

Example

log / tty / off

logadd

Add two logical blocks of coordinates.

Format

logadd / geometry / nx1, ny1, nz1 / first1 / nx2, ny2, nz2 / first2

logadd Parameters

Parameters	Type/Value	Description/Default
geometry	xyz	??
nx1, ny1, nz1	integers	
first1	real	
nx2, ny2, nz2	integers	
first2	real	

macro

Defines a macro to be used by the MACROUSE command for the MOVIE command.

The macro consists of three parts: name definition, macro body, terminator. When the command processor is in the macro definition mode then the command lines entered between the MACRODEF and MACROEND are not executed immediately but filed away for use when the macro is called for execution.

Format

macro

Example

macro

```
macrodef / movie1                    definition
  read / default              **
  viewread / view1              *  macro
  setvprt / 0.25, 0.25, 0.75, 0.75 *  body
  plotbob                     **
macroend                        terminator
```

macrodef

Defines the name of a macro. This name can be used to refer to this macro in the MACROUSE or MOVIE commands. This command indicates that we are starting the definition of a macro. A conditional macro is a macro with if-then blocks and is evaluated in the physics code. Conditional macros must start with an if-then statement and must end with endif. If statements are like fortran ifs except the logical operators are not deliminated with '.'s. Also the operands can only be global variables including the user set variables var0 through var9. (Note: The end of the macro is indicated by MACROEND.)

Format

macrodef / macro name

macrodef Parameters

Parameters	Type/Value	Description/Default
macro_name	char*32	

Example

```
macro
        macrodef / movie1                  definition
          read / default              **
          viewread / view1              *  macro
          setvprt / 0.25, 0.25, 0.75, 0.75 *  body
          pltbob                     **
        macroend                        terminator


    macro
        var1=.1                   set variable
        macrodef / cond1                  definition
          if ( time ge var1 ) then       **  conditional
            dtrs=.2                  *  macro
            var1=99999.                *  body
          endif                      **
        macroend                        terminator
```

macroend

Terminates a macro. This command closes a macro definition and returns to the normal interactive command processing.

Format

macroend

Example

```
macro
        macrodef / movie1                    definition
          read / default              **
          viewread / view1              *  macro
          setvprt / 0.25, 0.25, 0.75, 0.75 *  body
          pltbob                      **
        macroend                        terminator
```

macrouse

Executes a **macro** that has been defined with **macro** command.

Format

macrouse / macro name

macrouse Parameters

Parameters	Type/Value	Description/Default
macro_name	char*32	

Example

macrouse / macro1

mapgrid

Formats

mapgrid / ...

mapgrid Parameters

Parameters	Type/Value	Description/Default

mataddpt

Flags materials in which the user wishes to add mass points. Points are then added to tetrahedra in the flagged materials if the tetrahedra stretch too much in the direction of the flow.

Formats

mataddpt / material_options
mataddpt / allbut / list_of_materials_to_exclude

mataddpt Parameters

Parameters	Type/Value	Description/Default
material_option	List of material names or numbers	
	all	Selects all materials.
	0	Selects all materials.
	blank	Selects all materials.
allbut	allbut	Selects all except materials listed.
list of materials to exclude		

matchbdy

Join matching boundary faces together.

matrec

Adds materials to the material reconnection list. As the physics code runs, materials containing an overlap or merged points are automatically added to the list. Reconnecting flips will only take place in materials on the material reconnection list.

Format

matrec / material_list

matrec Parameters

Parameters	Type/Value	Description/Default
material_list	integers	mat1, mat2, ...matn
	all	

Example

matrec / air / water
matrec / all

memory

Estimates how much memory is needed to set up a problem, based on how many mass points will be used. Must be a GOOD guess.

Format

memory / number of mass points

memory Parameters

Parameters	Type/Value	Description/Default
number of mass_points	integer	Provide a good guess.

merge

Merge two points together

Format

merge / first_point / second_point

merge Parameters

Parameters	Type/Value	Description/Default
first_point, second_point	integer	On return, the first_point is the survivor unless first_point may not be removed (a corner point for example), then the command operates as if first_point and second_point have been interchanged. If there is no confirmation of the merge, one or both of the points may be inactive, or the merge may be illegal because the points are not neighbors or because this merge is disallowed by the merge tables.

Note

Merging may trigger other merges by the reconnection step that follows the merge.

Example

merge / 21, 22

modbld

Sets up the models for a particular problem (e.g., EOS, HE, etc.).

Format

modbld / part name / model name / type name / option

modbld Parameters

Parameters	Type/Value	Description/Default
part_name	char*32	
model_name	char*32	
type_name	char*32	
option		

Note

Mass points are associated with models with the **zq / imd** command.

Example

modbld / region1 / eos / sesame / eosid=3711,rho0=2.7, etc.

modchk

Checks the completness and consistancy of the MODELs that have been defined for a particular problem.

Format

modchk

Rules

This should be the first command after the models and points have been defined.

model

Sets the model of the grahics terminal that is being used.

Format

model / model

model Parameters

Parameters	Type/Value	Description/Default
model	4010	Tektronics 4010.
	4114	Tektronics 4114.
	4115	Tektronics 4115 with 8 planes. The default.
	41154	Tektronics 41154 with 4 planes.

modhelp

Checks the models for a particular problem (e.g., EOS, HE, etc.)

modpost

Prints a summary sheet for each model.

Format

modpost

modprt

Prints a summary sheet for each model.

Format

modprt

modset

Initializes the MODELs for a particular problem. At this point the SESAME tables are set up, and HE burn timing is computed.

Format

modset / model name / material list

modset Parameters

Parameters	Type/Value	Description/Default
model_name	char*32	Model name. Default is all.
material_list	char*32	List of material names or numbers to be processed. Default is all.

Rules

This command should follow the **modchk** command.

modval

An evaluation of all the PARTs of the current problem is done. After this command all the initial EOS and HE evalulations will be done.

Format

modval / model name / material list

modval Parameters

Parameters	Type/Value	Description/Default
model_name	char*32	Model name. Default is all.
material_list	char*32	List of materials to be processed. Default is all.

Rules

This command should follow the **modset** command.

mregion

Define a material region from a set of surfaces by logically combining the surface names and region names. The operators **lt, le, gt**, and **ge** are applied to previously defined surfaces according to the following rules.

Format

mregion / region_name / relation_oper surf_name [logic_oper relation_oper surf_name] […]

mregion Parameters

Parameters	Type/Value	Description/Default
region_name	char*32	Name of new region being defined by combing surfaces.

mregion Parameters

Parameters	Type/Value	Description/Default
relation_oper		The region is defined by logically combining the surface names with the surface commands. The parentheses operators, (and), are used for nesting. Spaces are required as delimiters to separate all operators and operands; parentheses are operators and must be surrounded by spaces. Internal interfaces should be excluded when defining material regions. (i.e. use **lt** and **gt**). External boundaries should be included when defining material regions. If a material regions consists of more than one region and the regions touch (i.e. share a region interface), then the region interface is not a material interface -- all the points on the region interface are interior to the material region. In this case use **le** or **ge** to include these region interface points in the material region as interior points.
	lt	If the surface following is a volume then **lt** means inside not including the surface of the volume. If the surface is a plane or a sheet **lt** means the space on the side of the plane or sheet opposite to the normal not including the plane or sheet itself.
	le	If the surface following is a volume then **le** means inside including the surface of the volume. If the surface is a plane or a sheet **le** means the space on the side of the plane or sheet opposite to the normal including the plane or sheet itself.
	gt	If the surface following is a volume then **gt** means outside not including the surface of the volume. If the surface is a plane or a sheet **gt** means the space on the same side of the plane or sheet as the normal not including the plane or sheet itself.
	ge	If the surface following is a volume then **ge** means outside including the surface of the volume. If the surface is a plane or a sheet **ge** means the space on the same side of the plane or sheet as the normal including the plane or sheet itself.
surf_name	char*32	Name of the surface combining.
logic_oper		Logical operators used to combine the relation_oper.

mregion Parameters

Parameters	Type/Value	Description/Default
	or	Union of surfaces.
	and	Intersection of surfaces.
	not	Complement of surface.

Rules

Interior interfaces must be included in one and only one region.

No two regions may overlap and there must be no voids.

The operators **lt**, **le**, **gt**, and **ge** can modify a previously defined surface, resulting in a regions.

The operators or, and, and not are applied to regions and mean union, intersection, and complement respectively.

The union of all regions must describe a connected object.

Parentheses are used for nesting. Parentheses are operators.

Spaces are required as delimiters to separate operators and operands.

Examples

mregion / mat1 / le box1 and (lt sphere1 and (lt plane1 or gt plane2))

mregion / mat2 / regiona or regionb

nb

Prints/Sets the number of reflective boundaries for a region.

Format

nb

newdict

Replaces the current parameter dictionary values with the ones from the default parameter dictionary.

Format

newdict

newmesh

Resets the generator so that another region can be defined.

Format

newmesh

nn

Prints information about the nearest neighbors of a given point.

Format

nn / point 1 / point 2

nn Parameters

Parameters	Type/Value	Description/Default
point_1	integer	Any mass point.
point_2	integer	Any mass point.

nn Point 1 Output

Value	Description
ntet	Number of tetrahedra associated with point 1.
mt	Equals imt
tp	Equals itp
hb	Equals ihb
rb	Equals irb
n1	Equals nn1
n2	Equals nn2

nn Point 1 Output

Value	Description
cr	Equals icr
nnj	Number of face neighbors for point 1.
nnk	Number of edge neighbors involved with all face neighbors.
nna	Array of face neighbors.
nn	Number of edge neighbors for each face neighbor.
x, y, z	coordinates
a	
v(point 1)	Sum of all partial volumes between point 1 and all face neighbors.
v(nna)	Sum of partial volumes between point 1 and each face neighbor.
nnb	Array of edge neighbors - listed as they are associated with each face neighbor.

nn Point 2 Output

Information in the output is the same as above except that:

- all values printed out apply to the nearest neighbors of both point 1 and point 2
- the connectivity between the two points is also printed out in the format described in the nntet command.

nnblock

Prints the vector block summary for the tetrahedrons associated with one, two, or three mass points. Included in this summary are the tetrahedron indexes and volumes.

Format

nnblock / n1 [/ n2 [/ n3]] [**reorder** / starting tet index]

nnblock Parameters

Parameters	Type/Value	Description/Default
n1, n2, n3		Mass points.
	n1	Prints a tet summary sheet. If only one tetrahedron is specified then the stored and calculated volume of the mass point is displayed.
	n1, n2	Prints the line summary sheet.
	n1, n2, n3	Prints the face summary sheet.
reorder	reorder	Reorder the tets to your specifications. (1 < tet index < number of tets)
Starting tet index	integer	

Examples

nnblock / n1

Prints a tet summary sheet.

nnblock / n1 / n2

Prints a line summary sheet.

nnblock / n1 / n2 / n3

Prints the face summary sheet.

nnblock / n1 / **reorder** / starting tet index.

nnblock / n1 / n2 / **reorder** / starting tet index.

nnblock / n1 / n2 / n3 / **reorder** / starting tet index.

nnsurf

This is the main command for generating the connectivity list.

Format

linkdan / isrchopt / ibndyopt / istpnt / 1st point, last point, stride / itetflag

nnsurf Parameters

Parameters	Type/Value	Description/Default
isrchopt	0	N-squared volume search.
	1	Same as 0 but includes additional closeness test.
	4	Iterative neighbor flipping to Delaunay mesh.
	5	N-squared surface search.
ibndyopt	0	Fill in boundary corners.
	1	Stop at first boundary situation encountered.
	2	Stop only when out of neighbors.
istpnt	integer	The relative point number to begin search with.
1st point, last point, stride	integers	Point sequence.
itetflag	0	Begin new tetrahedral arrays (overwrite existing arys).
	1	Add tets to the already existing tetrahedral arrays.

nntet

Describes all you could possibly want to know about a tetrahedron

Format

nntet / tet number

nntet Parameters

Parameters	Type/Value	Description/Default
tet_number	integer	

Helpful Information for the Output

- The four mass points (itet values) of the tet. (i1-i4)
- The position of the opposite points of i1-14 in the jtet array (j1-j4)
- The connectivity of the tet. (in order of appearance)

  1. face index of the specified point (It(1) <==> i1)
  2. tet. number containing the specified point
  3. face index of the opposite tet. (Jt(face index))
  4. number of the opposite tet.
  5. indexes of the generic tet. (i,j,k,l)
  6. index of the opposite vertex associated with the j-k-l face of the generic tet. (m). (j,k,l,m)=>opposite tet.

- Other values
- Vol = volume of tet.
- Dvol,T0,Dt0 = ?
- (xm,ym,zm) = centroid of tet.
- (x123,y123,z123) = centroid of face 123 (i1,i2,i3)
- A1-A4 = twice the area of the face opposite the point. (e.g. A1=2 x the area of face 234 (i2,i3,i4))
- Connections between points
- ds = distance between two points
- A = area of median mesh between two points
- Ap = ?
- (xa,ya,za) = vector normal to area A (1/2 magnitude)
- v1-v4 = partial volume of the tet. associated with two points (1/12 of the actual tet volume)

np

Prints a summary sheet on the point distribution for a region. The output includes the number of points in the region (**np** array - only the first element is used) and the number of tetrahedra (tets).

Format

np

offsetsurf

Offsets triangulated surfaces in the direction of the surface outward normal, i.e., normal surface motion. For each node a 'synthetic' unit outward normal **N** is computed based on the weighted average angle of the normals of the triangles shared by that node.

Format

offsetsurf / new_cmo / old_cmo

offsetsurf Parameters

Parameters	Type/Value	Description/Default
new_cmo	char*32	The name of the new surface.
old_cmo	char*32	The surface to be used in generating the offset surface.

Additional Information

To add the nodes in the new surface to the main mesh object use the **copypts** command. To add the new surface to the main mesh object use a **surface** command with new_cmo as the sheet name (e.g. **surface**/s2d/bndy_typedist is given in user coordinates (i.e. whatever units the old_cmo mesh object was defined in.) The new node coordinates, R_new, are computed using the formula:

r_new = r_old + dist * r_node

Examples

offsetsurf / cmo_out / cmo_in / d

offset a triangulated surface a distance d

offsetsurf / cmo_out / cmo_in / d / [xy, xz, yx, yz, zx, zy]

offset a line cmo a distance d in a direction parallel to the specified plane.

offsetsurf / cmo_out / cmo_in / d / x y z

offset a line cmo a distance d in the direction specified by the vector (x,y,z)

parallel

Invokes a parallel section of inputfile commands. The parallel section is terminated with a endparallel.

partition

Partitions a grid using METIS.

Format

partition / cmo_name / 1st partitions / 2nd partitions / [**elements** | **nodes**]

partition Parameters

Parameters	Type/Value	Description/Default
cmo_name	char*32	CMO name
1st partitions, 2nd partitions		The partition can be incremental where a grid can be partitioned into N-partitions, then these N-partiitons can be partitioned in M-partitions, so on. Inactive nodes or elements are spread among the partitions to balance the numbers/partition. Parents and children are put in the same partition.
elements | nodes	elements	Partitioning based on elements. This is the default.
	nodes	Partitioning based on nodes.

Examples

partition / cmo_name / 6 / 8

partition / -def- / 6 / 8
48 total partitions

partition / -def- / 0 / 128
48*128 total partitions

partition / -def- / 16 / 128 / nodes

partition / -def- / 16 / 128 / elements

perturb

Perturbs a point distribution according to SIN distributions. This is used to define spatial perturbations in stability calculations, such as the Young's Meshkov problem.

Format

perturb / 1st point, last point, stride / geometry / coordinate / xamp, yamp, zamp / xlam, ylam, zlam / xmin, ymin, zmin / xmax, ymax, zmax / xint, yint, zint / xcen, ycen, zcen

perturb Parameters

Parameters	Type/Value	Description/Default
1st point, last point, stride	integers	
geometry		
coordinate		
xamp, yamp, zamp	reals	
xlam, ylam, zlam	reals	
xmin, ymin, zmin, xmax, ymax, zmax	reals	
xint, yint, zint	reals	
xcen, ycen, zcen	reals	

plotreg

Plots a 3-d shaded image of selected regions or mregions. The default is to plot all regions.

Format

plotreg / ires / irtype / region_list | all | [allbut / ex_region_list]

plotreg Parameters

Parameters	Type/Value	Description/Default
ires		Resolution.
	l	Low with medium edge refinement. The default.
	lf	Low fast with no edge refinement
	m	Medium with fine edge refinement
	mf	Medium fast with no edge refinement
	h	High
irtype		Region type.
	region	
	mregion	The default.
region_list	char*32	List of region names to plot
all	all	Plot all regions.
allbut	allbut	Plot all regions, except those in ex_region_list.
ex_region_list	char*32	List of region names not to plot.

pregion

Saves, replaces, stores, or retrieves sets of **pstatus** points (also known as point regions).

Format

pregion / dir

pregion / save | replace | store / name / pstatus1, pstatus2, …, pstatusn / pstatus

pregion / get / name / array pointer

pregion Parameters

Parameters	Type/Value	Description/Default
	dir	Prints a directory of all point regions.
	save	The **pstatus** points with names pstatus1 through pstatusn are saved under point region name.
	replace	The **pstatus** points in point region name are replaced by the **pstatus** points with names pstatus1 through pstatusn.
	store	The **pstatus** points in point region name are overwritten by the **pstatus** points with names pstatus1 through pstatusn.
	get	Retrieves the set of **pstatus** points in point region name.
name	char*32	Name of region points.
pstatus1, pstatus2, …, pstatusn		
pstatus		
array pointer		An array pointer used for the array containing the mesh size for the point region.

process

Assigns the processee and processor numbers that run this job.

pset

Give a name to a selected set of points.

Formats

pset / pset_name **/ seq** / ifirst, ilast, istride

pset / pset_name / logic_oper **/** pset1 [pset2 / ... / psetn]
pset / pset_name / **list**
pset / pset_name / **delete**
pset / pset_name / **attribute** / attribute_name / ifirst, ilast, istride / value / relation_oper
pset / pset_name / **zq** / attribute_name / ifirst, ilast, istride / value / relation_oper
pset / pset_name / **region** / region_name /ifirst, ilast, istride
pset / pset_name / **mregion** / mregion name / ifirst, ilast, istride
pset / pset_name **/ geom.** / **xyz** / ifirst, ilast, istride / xl, yl, zl / xu, yu, zu / xcen, ycen, zcen
pset / pset_name / **geom.** / **rtz** / ifirst, ilast, istride / r1, t1, z1 / r2, t2, z2 / xcen, ycen, zcen
pset / pset_name / **geom.** / **rtp** / ifirst, ilast, istride / r1, t1, p1 / r2, t2, p2 / xcen, ycen, zcen
pset / pset_name / **surface** / surface_name / [ifirst, ilast, istride]
pset / pset_name / **eset** / eset_name

pset Parameters

Parameters	Type/Value	Description/Default
pset_name	char*32	
seq	seq	forms a pset of the nodes in ifirst, ilast, istride
ifirst, ilast, istride	integers	1,0,0 refers to all nodes; 0,0,0 refers to the last set of nodes created; i,j,k refers to points I through j using stride k.
logic_oper	union inter not delete	Logical operations on previously defined psets. The definition of the unary operator **not** is extended such that **not**/p1/p2 means p1 and (not(p2)).
pset1 [pset2 / ... / psetn	char*32	
list	list	**list** lists nodes in a pset or names of all psets
delete	delete	**delete** deletes a previously defined pset

pset Parameters

Parameters	Type/Value	Description/Default
attribute	attribute	**attribute** forms a pset from all points in ifirst,ilast,istride which have the specified value for a node based attribute. This option was previously named **zq.** If the optional comparator field is given; that operation is used to compare the attribute value to the requested value.
attribute_name	char*32	
value	eal	
relation_oper	le ge lt le ne eq	
zq	zq	See attribute above, under this command.
region	region	
region_name	char*32	
mregion	mregion	Will return all nodes that are in the specified mregion - the definition of the mregion is evaluated to determine membership. Hence the result may vary from what would be returned if the 'imt1' value of the nodes had been queried using the **attribute** option.
mregion_name	char*32	
geom	geom	
xyz	xyz	Forms a pset from all points inside the box whose corners are xl,yl,zl and xu,yu,zu relative to the geometry center at xc,yc,zc.
xl, yl, zl	reals	
xu, yu, zu	reals	

pset Parameters

Parameters	Type/Value	Description/Default
xcen, ycen, zcen	reals	
rtz	rtz	Forms a pset of nodes within the cylinder or cylindrical shell given by radius r1 to r2, angle t1 to t2 and height z1 to z2.
r1, t1, z1	reals	
rtp	rtp	Forms a pset of nodes within the sphere, sperical shell or sperical section given by r1 to r2, and angles t1 to t2 and p1 to p2. See the section of Mesh Definitions for an explanation of angles theta and phi.
r1, t1, p1	reals	
r2, t2, p2	reals	
surface	surface	Selects the nodes that are on the specified surface.
surface_name	-interface-	All interface nodes will be returned.
	-boundary-	All nodes on external bounding surfaces will be returned.
	-all-	All nodes on any surface are returned.
eset	eset	Return all the nodes that are vertices of the elements in the specified element set.
eset_name	char*32	Previously assigned element set name

Examples

pset / apset / **seq** /1, 0, 0

associate the pset name apset with all points.

pset / apset / **seq** / 0, 0, 0

associate the pset name apset with the last set of nodes created.

pset / apset / **union** / pset1, pset2, pset3

associate the pset name apset with the set of nodes which belong to at least one of pset1, pset2, pset3.

pset / apset / **inter** / pset1, pset2, pset3

associate the pset name apset with the set of nodes which belong to pset1, and pset2, and pset3.

pset / apset / **not** / pset1, pset2, pset3

associate the pset name apset with the set of nodes which belong to pset1, and do not belong to pset2, and do not belong to pset3

pset / apset / **not** / pset1

associate the pset name apset with the set of nodes which do not belong to pset1

pset / / **list**

list the names of all psets

pset / apset / **attribute** / **itp** / 1, 0, 0 / 10 / **ge**

associate the name apset with the points whose type field (**itp1**) has value greater than or equal to 10 (these would be boundary nodes).

pset / mypset / **list**

list the node numbers of the members of mypset

pset / mypset / **geom.** / **xyz** / 1, 0, 0 / 1., 1., -5. / 10., 20., 10.

associate the name mypset with all nodes that fall with the box with corners at (1,1,-5) and (10,20,10)

pset / mypset / **geom. / rtz / pset, get,** oldpset

associate the name spseta with the set of nodes that lie on the surface s2 and which are members of the pset spset This command and the previous command would identify the nodes that are on the intersection of surfaces s1 and s2 and give the name spseta to these nodes.

Assigns the processee and processor numbers that run this job.

pstatus

Saves, removes, retrieves, or replaces a specified set of points, usually the last set of points defined by a generator command or the set of points defined by ifirst,ilast,istride. Note that point sets must be specified in sequence in order for this command to work properly.

Format

pstatus
pstatus / **save** | **store** / name / ifirst, ilast, istride
pstatus / delete | **get /** name

pstatus Parameters

Parameters	Type/Value	Description/Default
2nd argument	blank	Returns current point status counters.
	save	Saves the point status numbers, ifirst,ilast,istride under name.
	store	Overwrites what was in name with ifirst,ilast,istride
	delete	Deletes values from name
	get	Retrieves values from name
name		
ifirst, ilast, istride		

pyoungs

Perturbs a point distribution according to SIN distributions. This is used to define spatial perturbations in stability calculations, such as the Young's Meshkov problem.

Format

perturb / 1st point, last point, stride / geometry / coordinate / xamp, yamp, zamp / xlam, ylam, zlam / xmin, ymin, zmin / xmax, ymax, zmax / xint, yint, zint / xcen, ycen, zcen

pyoungs Parameters

Parameters	Type/Value	Description/Default
1st point, last point, stride	integers	
geometry		

pyoungs Parameters

Parameters	Type/Value	Description/Default
coordinate		
xamp, yamp, zamp	reals	
xlam, ylam, zlam	reals	
xmin, ymin, zmin, xmax, ymax, zmax	reals	
xint, yint, zint	reals	
xcen, ycen, zcen	reals	

quadxy

Define an arbitrary, logical quad of points in 2D(xy) space.

Format

quadxy / nx , ny / x1, y1, z1 / x2, y2, z2 / x3, y3, z3 / x4, y4, z4

quadxy Parameters

Parameters	Type/Value	Description/Default
nx, ny	integer	Specify the number of points in the x and y directions.
x1, y1, z1 x2, y2, z2 x3, y3, z3 x4, y4, z4	reals	The four corners of the quad are listed in counter clockwise order (the normal to the quad points is defined using the right hand rule and the order of the points).

quadxyz

Define an arbitrary, logical hexahedron of points in 3D(xyz) space.

Format

quadxyz / nx, ny, nz / x1, y1, z1 / x2, y2, z2 / x3, y3, z3 / x4, y4, z4 / x5, y5, z5 / x6, y6, z6 / x7, y7, z7 / x8, y8, z8

quadxyz Parameters

Parameters	Type/Value	Description/Default
nx	integer	Specifies the number of points between the 1st and 2nd point, the x points.
ny	integer	Specifies the number of points between the 1st and 4th point, the y points.
nz	integer	Specifies the number of points between the 1st and 5th point, the z points.
x1, y1, z1 thru x8, y8, z8	reals	The eight corners of the hex are listed as two sets of quads, each set of four nodes is given in counter clockwise order . Points 1 to 4 specify one face of the hex, points 5 to 8 the corresponding face opposite (point 5 is logically behind point 1, point 6 behind point 2 and so on.)

Examples

quadxyz / 5 25 7 / 0. 0. 0. / 1. 0. 0. / 1. .0 10. / 0. .0 10. / 0. 2.0 0. / 1. 2. 0. / 1.0 2. 10.0 / 0. 2. 10.0

quadxyz /5 25 7 / 0. 0. 0. / 1. 0. 0. / 1. 2. 0. / 0. 2. 0. / 0. 0. 10. / 1. 0. 10. / 1. 2. 10. / 0. 2. 10.

random

Randomizes a set of points within a set of restrictions.

Format

random / geometry / 1st point, last point, stride / xmin,ymin,zmin / xmax, ymax, zmax / dx, dy, dz

random Parameters

Parameters	Type/Value	Description/Default
geometry	xyz	Specifies Cartesian coordinates.
1st point, last point, stride	integer	Set of restrictions based on point number limits.
xmin, ymin, zmin xmax, ymax, zmax	reals	Set of restrictions based on physical limits.
dx, dy, dz	reals	Represent the degree of variation in each direction. The larger the variation, the higher degree of randomness.

read

This command reads in data into the active Mesh Object, replacing whatever data might have been previously contained in the active Mesh Object.

Format

read / file_type / file_name

read Parameters

Parameters	Type/Value	Description/Default
file_type	**x3d**	Will restart the code at the state at which the dump command was issued.

read Parameters

Parameters	Type/Value	Description/Default
	avs **datex** **dcm** **free** **gmv** **iges_grid** **ngp** **rage** **stl** **vrml** **xdl**	
file_name	char*32	

Example

read / **x3d** / myfile

recon

This command flips connections in the mesh to get restore the Delaunay criterion or to improve element shapes.

Format

recon / [**1** | **0**] / [damage]

recon Parameters

Parameters	Type/Value	Description/Default
	1	Adds points on the boundaries if needed. recommended for 2D meshes only
	0	Specifies that no points are to be added on the boundaries. This is the default.

recon Parameters

Parameters	Type/Value	Description/Default
damage	damage	If specified then flips on exterior boundaries are checked to verify that the maximum depth of deformation of the external boundary does not exceed the value of damage. The default value of damage is 1% of the problem size. This setting prevents connecting across corners if the external boundary is a reflective box.

Additional Flipping Criteria

The standard method used by recon is to flip connections based on the in-sphere test (the circumsphere of a tetrahedral element should contain no other nodes). Additional flipping criteria are available. The Minimum Error Gradient Adaption (mega) can be invoked by changing the value of the code variable ivoronoi(**assign**///ivoronoi/-2). The effect of this option is to generate well shaped elements; however the grid will not be Delaunay. If the user has a function to used for adaptive reconnection this option is available by setting the code variable ivoronoi to 2 (**assign**///ivoronoi/2). The user will have to supply an external function.

Examples

recon

attempt to restore Delaunay

assign / / / ivoronoi / -2

recon

attempt to improve geometric mesh quality

recon / **1**

for 2d meshes add nodes on boundaries to guarantee Delaunay

recon / / .001

reconnect limit interface and boundary damage to a maximum of .001.

refine

The **refine** command is used to create more elements when a criterion is specified.

Format

refine / refine_criterion / field_type / interpolation / refine_type

refine Parameters

Parameters	Type/Value	Description/Default
refine_criterion	junction	Wll refine object where field crosses xrefine.
	roughness	Wll refine based on the distance of the endpoint of an edge to the plane determinded by the synthetic normal with respect to a specified surface at the other endpoint of the edge.
	edge_list	Will bisect a set of edges specified by the node numbers of the endpoints of the edges.
	interface	Will bisect a set of noninterface edges of tets all of whose vertices are interface nodes.
	maxsize	Wll refine object where object > xrefine. **maxsize** refers to volume for tets, area for face, length for edges
	constant	Wll refine object where field > xrefine.
	addpts	Will refine explicitly by adding in a set of nodes.
	rivara	Edges longer than xrefine will be refined according to the principle that a refined edge candidate is the longest edge in any element that contains it. This results in a recursive refinement procedure that adds neighboring edges to the refinement candidate list until no neighbor edge is longer then the candidate. Refine_type must be **edge**. Field and interpolation are ignored.
	rvara_boundary	Apply the rivara algorithm, but only bisect edges on external boundaries. **rivera_truncated**
	rvara_truncated	Apply the rivara algorithm, but restrict the neighborhood search to the edges in the input pset. If the pset is the entire mesh this option has the same behavior as **rivara**.
	delta	Will refine object where delta(field) > xrefine.
	lambda	Wll refine object where lambda (field) < xrefine.

refine Parameters

Parameters	Type/Value	Description/Default
field_type	char*32	The field must refer to a previously defined attribute of the current Mesh Object.
interpolation		The interpolation specifies how to interpolate the field to give field values to the new nodes created.
	linear	
	log	
	asinh	
refine_type		Secifies what object will be refined and how that object will be refined.
	tet	Will refine elements by placing a point in the center of the element.
	face	Will refine facets by placing a point in the center of the facet.
	edge	Will refine edges by placing a point on the midpoint of the edge.
	faceedge	Will refine facets by refining all edges of the facet.
	tetedge	Will refine elements by refining all edges of the element.
inclusion_flag		Specifies if refinement is an inclusive or an exclusive operation.
	inclusive	If for example, an edge refinement is specified restricted to a **pset**, then an edge is eligible for refinement if either or both of the end points belong to the **pset** if the inclusion_flag is set to **inclusive**.
	exclusive	If the inclusion_flag is **exclusive** then both end points must be in the **pset**.

Specific examples by refine_criterion

refine / roughness / / / edge / ifirst, ilast, istride / distance / surface_name / **exclusive | inclusive**

refine / edge_list / / / edge / edge_list

refine / interface / / / edge / pset, get, psetname / / / / [**inclusive** | **exclusive**]
refine / maxsize / / / edge / pset, get, something / .25

refine/**constant**/concentration/**log**/**edge**/ 1,0,0/25.0///**inclusive**

refine/**addpts**///**tet**/**pset,get,**newpoints/

refine/**rivara**///**edge/pset/get,**p1/.5///**inclusive**
refine all edges containing at least one node in pset p1 that are longer than .5. Using the 'rivera' algorithm may result in edges not containing nodes in the pset to be refined.

refine/rivera_truncated///edge/pset,get,p1/.5///**exclusive**
rivera_truncated, exclusive will refine only edges both of whose endpoints are in the pset

refine/rivara_boundary///edge/1,0,0/.25
rivara_boundary will only refine boundary edges.

refine/roughness///edge/1,0,0/.28/ptop**/inclusive**

refine/edge_list///edge/1 2 23 47/
will refine the edge with endpoints 1 and 2 AND the edge with endpoints 23 and 47.

region

The **region** command defines a geometric region from the set of surfaces by logically combining the surface names.

Format

region / region_name / relation_oper surf_name [logic_oper relation_oper surf_name] […]

region Parameters

Parameters	Type/Value	Description/Default
region_name	char*32	Name of new region being defined by combing surfaces.

region Parameters

Parameters	Type/Value	Description/Default
relation_oper		The region is defined by logically combining the surface names with the surface commands. The parentheses operators, (and), are used for nesting. Spaces are required as delimiters to separate all operators and operands. Internal interfaces should be included in exactly one region. In the event of conflicting region commands, the one occurring last in the input stream takes precedence.
	lt	If the surface following is a volume then **lt** means inside not including the surface of the volume. If the surface is a plane or a sheet **lt** means the space on the side of the plane or sheet opposite to the normal not including the plane or sheet itself.
	le	If the surface following is a volume then **le** means inside including the surface of the volume. If the surface is a plane or a sheet **le** means the space on the side of the plane or sheet opposite to the normal including the plane or sheet itself.
	gt	If the surface following is a volume then **gt** means outside not including the surface of the volume. If the surface is a plane or a sheet **gt** means the space on the same side of the plane or sheet as the normal not including the plane or sheet itself.
	ge	If the surface following is a volume then **ge** means outside including the surface of the volume. If the surface is a plane or a sheet **ge** means the space on the same side of the plane or sheet as the normal including the plane or sheet itself.
surf_name	char*32	Name of the surface combining.
logic_oper		Logical operators used to combine the relation_oper.
	or	Union of surfaces.
	and	Intersection of surfaces.
	not	Complement of surface.

Rules

Interior interfaces must be included in one and only one region.

No two regions may overlap and there must be no voids.

The operators **lt**, **le**, **gt**, and **ge** can modify a previously defined surface, resulting in a regions.

The operators or, and, and not are applied to regions and mean union, intersection, and complement respectively.

The union of all regions must describe a connected object.

Parentheses are used for nesting. Parentheses are operators.

Spaces are required as delimiters to separate operators and operands.

Examples

region / reg1 / **le** sphere1 **and** (**lt** plane1 **or gt** plane2)
region / reg2 / **le** sphere1 **and** (**ge** plane1 **and le** plane2)

regnpts

Generates points in a region previously defined by the region command. The points are generated by shooting rays through a user specified set of points from an origin point, line, or plane and finding the intersection of each ray with the surfaces that define the region.

Format

regnpts / region_name / ptdist / ifirst, ilast, istride / geom / ray_origin / irratio, rrz / maxpenetr
regnpts / region_name / ptdist / **pset**, **get**, setname / geom / ray_origin / irratio, rrz / maxpenetr

regnpts Parameters

Parameters	Type/Value	Description/Default
region_name	char*32	
ptdist		The point distribution is determined by the data in ptdist.
	integer	That many points are evenly distributed along the ray in the region. The variables **irratio** and **rrz** determine ratio zoning when ptdist is an integer.

regnpts Parameters

Parameters	Type/Value	Description/Default
	real	Points are distributed at that distance along the ray, up to a maximum of **maxpenetr** points along the ray (in addition to any interface points that may be created). note: If the ray encounters a region more than once, multiple sets of points are layed down. Points are distributed on the region ís material interfaces and external boundaries if the region definition includes the interfaces or boundaries -- usually **ge** or **le** means that the region includes the interface or boundary.
	in inside out outside both	Only surface intersection points are created if ptdist is **inside**, **in**, **out**, **outside**, or **both.** In this case, surface points are created regardless of region ownership of the interface or boundary surface -- if a ray encounters a region more than once, the appropriate surface intersection point(s) is generated for each encounter. If another region intrudes upon the **regnpts** region so that the **regnpts** region is divided into more than one piece, points that fall inside the intruding region are not distributed.
ifirst, ilast, istride	integers	Define the set of points to shoot rays through.
geom	xyz	Points 1, 2, 3 define the plane to shoot rays from that are normal to the plane.
	rtz	Points 1, 2 define the line from which to shoot perpendicular rays.
	rtp	**xcen, ycen,** and **zcen** define a point from which to shoot rays.
	points	2 point sets have the same length and rays are constructed between pairs of elements, one from each point set.
ray_origin		
	x1, y1, z1 / x2, y2, z2 / x3, y3, z3	Used with **xyz** geom.

regnpts Parameters

Parameters	Type/Value	Description/Default
	x1, y1, z1 / x2, y2, z2	Used with **rtz** geom.
	xcen, ycen, zcen	Used with **rtp** geom.
	iffirst,iflast,ifstride	Used with points geom.
irratio, rrz		If **ptdist** is an integer, **irratio** and **rrz** determine the ratio zoning. If **irratio** =1, ratio zoning is on and the point distribution is adjusted so that the ratio between successive pairs of points is **rrz** If **irratio** = 3, ratio zoning is calculated on the longest ray; then this length distribution is applied to all rays. See the description of the command **surface** for a discussion of point distributions with respect to sheet surfaces.
maxpenetr		If **ptdist** is a real, it can distribute up to a maximum of **maxpenetr** points along the ray.
pset get pset_name		Define the set of points to shoot rays through. See **pset** command.

Rules

The points and the plane must lie outside the enclosing volume and on opposite sides.

The normal to the plane must point toward the point.

As rays are created, if they do not pass through the specified region, no points are distributed.

Points may be spaces evenly along the ray or they may be spaced according to a ration.

Specific Geometric Formats

regnpts / region_name / ptdist / ifirst, ilast, istride / **xyz** / x1, y1, z1 / x2, y2, z2 / x3, y3, z3 / irratio, rrz / maxpenetr

Where points 1, 2, 3 define the plane to shoot rays from that are normal to the plane.

regnpts / region_name / ptdist / ifirst, ilast, istride / **rtz** / x1, y1, z1 / x2, y2, z2 / irratio, rrz

Where points 1, 2, define the line from which to shoot perpendicular rays

regnpts / region_name / ptdist / ifirst, ilast, istride / **rtp** / xcen, ycen, zcen / irratio, rrz / maxpenetr

Where xcen,ycen,zcen define a point from which to shoot rays .

regnpts / region name / ptdist / ifirst, ilast, istride / **points** /iffirst, iflast, ifstride / irratio, rrz / maxpenetr

Where the 2 point sets have the same length and rays are constructed between pairs of elements, one from each point set.

Examples

resetpts

Reset node values. If iflag is **parent** (default) the parent child flags are reset. All child points are eliminated and the connectivity list is corrected to reference only the parent points. If iflag is **itp** the itp1 array is reset to indicate whether each node is in the interior (0), on an interior interface (2), on a reflected boundary (10), or on a reflected interface boundry (12) . Resetting itp would be used if nodes were removed (such as with rmmat) leaving new boundaries

Formats

resetpts / iflag
resetpts / cell_color / integer_node_color
resetpts / cell_color / [arg1 arg2 arg3]

resetpts Parameters

Parameters	Type/Value	Description/Default
iflag	integer	
cell_color		
integer_node _color	imt	
arg1, arg2, arg3	integers	For the case where there are no arguments or three arguments after cell_color the result is: Loop through all itetclr values 1 to maxclr, and reset node imt to element colors, **itetclr**. Istart, istride, iend can be used to loop subset of element colors(itetclr).

Note

This option loops through each node of each element and if the node color (imt) is equal to the user specified value (integer_node_color) then it is changed to the element color (itetclr). This will introduce a bias since the nodes are modified in the order of the element numbering. To give some control over the bias the user can specify a negative value for integer_node_color. In that case, the element loop goes from largest to smallest element number.

Examples

resetpts
resetpts / parent
resetpts / itp
resetpts / cell_color / 1

resetpts / cell_color
loop through colors from itetclr_min to itetclr_max

resetpts / cell_color / 1, 0, 0
loop through colors from itetclr_min to itetclr_max

resetpts / cell_color / 1, 3, 1
loop through colors from itetclr=1 to itetclr=3

resetpts / cell_color / 3, 1, -1
loop through colors from itetclr=3 to itetclr=1

restart

Restarts the code with a new dump file

Format

restart / file name / dump name / region name

restart Parameters

Parameters	Type/Value	Description/Default
file_name	char*32	
dump_name	char*32	
region_name	char*32	

rm

Removes any points that are within the specified point range and specified volume of space.

Format

rm / geom / ifirst, ilast, istride / xmin, ymin, zmin / xmax, ymax, zmax / xcen, ycen, zcen / [xscale, yscale, zscale]

rm Parameters

Parameters	Type/Value	Description/Default
geom	xyz	Cartesian coordinates.
	rtz	Cylindrical coordinates. theta is the angle in the XY-plane with respect to the x-axis. The cylinder always lines up along the z axis.
	rtp	Spherical coordinates. theta is the angle with respect to the Z-axis and phi is the angle in the XY-plane with respect to the X-axis.
ifirst, ilast, istride	integers	Point range to search.
xmin, ymin, zmin	reals	Minimums of geometry type coordinates.
xmax, ymax, zmax	reals	Maximums of geometry type coordinates.
xcen, ycen, zcen	reals	Center of removal space for geometry.
xscale, yscale, zscale	reals	Scaling factors for geometry limits.

Note

Use the **coordsys** command before issuing the **rm** command if the points to be removed are not aligned with the z-axis; then issue a final **coordsys** command to return to normal. Also note that the points that are removed become dudded out (point type set to 21) and not removed from the data array.

Example

rm / **xyz** / 1, 0, 0 / 2., 2., 2. / 4., 4., 4. / 0., 0., 0.
rm / rtz / 1, 0, 0 / 0., 0., 0. / 1., 360., 10. / 0., 0., 0.

rmmat

This routine is used to remove points that are of a specified material type. Elements with the specified material types are flagged by setting the element material type negative. After using **rmmat**, **rmpoints/compress** will delete elements whose material type is negative and the dudded nodes.

Format

rmmat / material number / **[all | node | element] / [exclusive]**

rmmat Parameters

Parameters	Type/Value	Description/Default
material_number		
all | node | element	all	Removes nodes with imt = material number and removes elements with itetclr= material number. This is the default.
	node	Removes nodes with imt = material number
	element	Removes elements with itetclr= material number
exclusive	exclusive	Removes everything except nodes with imt =material and removes everything except elements with itetclr= material.

Note

After using **rmmat**, **rmpoints / compress** will delete elements whose material type is negative and the dudded nodes.

rmpoint

Removes nodes or marks nodes for removal points or removes elements from a mesh.

Formats

rmpoint / ifirst, ilast, istride / [**exclusive** | **inclusive**]
rmpoint / **compress**
rmpoint / zero_volume / threshold
rmpoint / element
rmpoint / element / tet list
rmpoint / element / eset, get, esetname

rmpoint Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride		Sets the node type flag [itp=**ifitpdud** (21)] to indicate that the set of nodes are treated as invisible, but does not actually remove the nodes.
exclusive | inclusive	exclusive	Any element containing a retained node is retained. Is the default.
	inclusive	Any element containing a marked node will be removed.
compress	compress	Removes the invisible nodes (i.e. those nodes whose itp1 value is 21) from the data structure and material-wise resequences all remaining nodes.
zero_volume	zero_volum e	Remove elements whose volumes are less than or equal to the specified threshold.
threshold	real	
element		Removes all marked elements from the mesh. Marked elements have a negative value for the first entry in the itet vertex list.
tet_list	integers	List of elements.
eset, get, esetname		Named element set.

Example

rmpoint / pset, get, pset1

mark all the nodes in pset1 for removal. Remove elements all of whose vertices are members of pset1.

rmpoint / compress

remove all marked nodes and correct the itet array

rmpoint / zero_volume / 1.e-16
remove all elements with volumes less than 1.e-16

rmpoint / element / 27 259 1009
remove the three specified elements

rmpoint / element / eltset, get, myeset

rmregion

Removes points that lie within the specified region.

Format

rmregion / region_name

rmregion Parameters

Parameters	Type/Value	Description/Default
region_name	char*32	

rmsphere

Removes a sphere of points from a point distribution.

Format

rmsphere / inner_radius / outer_radius / xcen, ycen, zcen

rmsphere Parameters

Parameters	Type/Value	Description/Default
inner_radius	real	
outer_radius	real	

rmsphere Parameters

Parameters	Type/Value	Description/Default
xcen, ycen, zcen	reals	

rmsurf

Removes points that lie in, on or in and on the specified surface.

Format

rmsurf / surface_name / ioper

rmsurf Parameters

Parameters	Type/Value	Description/Default
surface_name	char*32	
ioper	lt	Only points in the surface are removed.
	eq	Only points on the surface are removed.
	le	All points in or on the surface are removed.

rotateln

Rotates a point distribution about a line.

Format

rotateln / ifirst, ilast, istride / **nocopy** | **copy** / x1, y1, z1 / x2, y2, z2 / theta / xcen, ycen, zcen/

rotateln Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride		Defines the point distribution.

rotateln Parameters

Parameters	Type/Value	Description/Default
nocopy | copy	nocopy	Just keeps the rotated points themselves.
	copy	Makes a copy of the original points as well as the rotated points.
x1, y1, z1 x2, yx, z2		Line of rotation defined by x1 through z2. Needs to be defined such that the endpoints extend beyond the point distribution being rotated.
theta	real	The angle of rotation, in degrees, whose positive direction is determined by the right-hand-rule, that is, if the thumb of your right hand points in the direction of the line (1 to 2), then your fingers will curl in the direction of rotation.
xcen, ycen, zcen	reals	The point where the line can be shifted to before rotation takes place.

Note

The end points of the line segment must extend well beyond the point set being rotated.

Example

rotatept

Rotates a point distribution about a point.

Format

rotatept / first, ilast, istride / **nocopy | copy** / xcen, ycen, zcen / theta / phi

rotatept Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride	integer	Defines the point distribution.
nocopy | copy	nocopy	Just keeps the rotated points themselves.

rotatept Parameters

Parameters	Type/Value	Description/Default
	copy	Makes a copy of the original points as well as the rotated points.
xcen, ycen, zcen	reals	Point to rotate point distribution about.
theta	real	The angle of rotation, in degrees, of the xy plane around the Z-axis, where positive phi is measured from the positive x-axis toward the positive y-axis.
phi	integer	The angle of rotation, in degrees, toward the negative z-axis.

rz

This command adds points to the mesh. It can distribute points evenly or according to a ratio zoning method.

Format

rz / geom | ni, nj, nk / xmin, ymin, zmin / xmax, ymax, zmax / iiz, ijz, ikz / [iirat, ijrat, ikrat / xrz, yrz, zrz]

rz / line / np / / / xmin, ymin, zmin, xmax, ymax, zmax / iiz, ijz, ikz

rz Parameters

Parameters	Type/Value	Description/Default
geom	xyz	Specifies Cartesian coordinates.
	rtz	Specifies cylindrical coordinates. The center is at (0,0,0). Use a **trans** command to move the center. The minimum and maximum coordinates are the triplets: radius from the cylinder's axis, angle in the xy-plane measured from the x-axis and height along the z-axis. Note that the **rtz** always results in a (partial) cylinder of points centered around the z axis. Use the **rotateln** command to orient the cylinder. For example, to center the cylinder around the y axis, specify the x axis as the line of rotation in the **rotateln** command.

rz Parameters

Parameters	Type/Value	Description/Default
	rtp	Specifies spherical coordinates. The center is at (0,0,0). Use a **trans** command to move the center. The command minimum and maximum coordinates are the triplets: radius from the center of the sphere axis, angle in the zy-plane measured from the positive z-axis and the angle in the xy-plane measured from the positive x-axis (see section on mesh definitions).
ni, nj, nk	integers	Number of points to be created in each direction.
xmin, ymin, zmin	reals	Minimums for coordinates.
xmax, ymax, zmax	reals	Maximums for coordinates.
iiz, ijz, ikz	integers	if =0 then mins and maxs are used as cell centers if =1 then mins and maxs are used as cell vertices
iirat, ijrat, ikrat	integers	ratio zoning switches (0=off,1=on)
xrz, yrz, zrz	reals	Ratio zoning value - distance is multiplied by this value for each subsequent point.
line	line	This option implies xyz and will distribute n1 nodes from (xmin,ymin,zmin) to (xmax,ymax,zmax).
np	integer	

Note

Filter should be used to remove these duplicate points.

Examples

rz / xyz / 5, 3, 10 / 0., 2., 0. / 5., 6., 2. / 1, 1, 1

This results in a set of 150 points, five across from x=0. to x=5., 3 deep from y=2. to y=6. and 10 high from z=0. to z=2.

rz / rtz / 4, 6, 11 / 0., 0., 0. / 3., 360., 10. / 1, 0, 1

This results in 264 points arranged around the z- axis. There are 3 rings of points at distances r=1., r=2. and r=3. from the z-axis. There are 11 sets of these three rings of points and heights z=0., z=1., z=2.,...,z=10. In each ring there are 6 points where

each pair of points is separated by 60°; note that ijz=0 requests that points be placed at cell centers, hence the first point will be at 30° not at 0°. Corresponding to r=0, there will be 6 identical points at 11 intervals along the z-axis at heights z=0., z=1., z=2.,...z=10.

rzbrick

Builds a brick mesh and generates a nearest neighbor connectivity matrix. This command is similar to the **rz** command format except here we have symmetry flags to input. A second format specifies that a mesh be created and connected.

Format

rzbrick / geom / ni, nj, nk / xmin, ymin, zmin / xmax, ymax, zmax / iiz, ijz, ikz / [iirat, ijrat, ikrat / xrz, yrz, zrz / isym, jsym, ksym]

rzbrick / geom **/** ni, nj, nk / **pset, get,** name / **connect**

rzbrick / interp / relative / 0 / **pset, get**, p1 /, p2 / cmo_new [/ pset, get, pset_path / pset, get, pset_rotation pset, get

rzbrick Parameters

Parameters	Type/Value	Description/Default
geom	xyz	Specifies Cartesian coordinates.
	rtz	Specifies cylindrical coordinates.
	rtp	Specifies spherical coordinates.
ni, nj, nk	integers	Number of points to be created in each direction.
xmin, ymin, zmin	reals	Minimums for coordinates.
xmax, ymax, zmax	reals	Maximums for coordinates.
iiz, ijz, ikz	0, 0, 0	mins and maxs are used as cell centers.
	1, 1, 1	mins and maxs are used as cell vertices.
iirat, ijrat, ikrat		Ratio zoning switches.
	0	off

rzbrick Parameters

Parameters	Type/Value	Description/Default
	1	on
xrz, yrz, zrz		Ratio zoning value - distance is multiplied by the value for each subsequent point.
isym, jsym, ksym		Symmetry flags.
pset, get name		Starting point
connect	connect	Use this option with **quadxyz** to connect logically rectangular grids.

Example

rzbrick / xyz / 3, 2, 3 / 0., 0., 0. / 1., 1. ,1. / 1, 1 ,1
creates a hex grid 2x1x2 cells in the unit cube
quadxyz / 5, 7, 5 / 0., 0., 0. / 1., 0. ,0. / 1.5, 0.5, 2.0 / .5, .2, 2.5 / -1., 1.5, 0. / 2. 0, 0., 0.0 / 2.1, 1.9, 2.4 / -0.2, 1.8, 2.3
setpts
rzbrick / xyz / 5, 7, 5 / 1 ,0, 0 / **connect**
creates a hex grid inside the hexahedral specified by the 8 corners passed to **quadxyz**

rzd

Refines a contour or generates points between two contours.

Formats

rzd / **1d** / nr / ii1, ii2, ii3
rzd / **2dxy0 | 2dxy1** / nr / ii1, ii2, ii3 / jj1, jj2, jj3 / xcen, ycen, zcen / irz / irratio / rrz

rzd Parameters

Parameters	Type/Value	Description/Default
1d	1d	Refines a given contour (specified by ii1,ii2,ii3) by placing a number (nr) of points between each pair of points on the contour.

rzd Parameters

Parameters	Type/Value	Description/Default
nr		Number of points. Note, this command works best if both contours have equal number of points.
	integer	Specifies the number of points to be placed between each set of matched points.
	real	Specifies the distance between points placed between each set of matched points.
ii1, ii2, ii3	integers	1st point, last point, stride of the first contour. This can be specified by using **pset, get**, setname, where setname is the sequence of points that has been set earlier using the **pset** command.
2dxy0	2dxy0	Generates points between two contours (ii1, ii2, ii3) and (jj1, jj2, jj3) with a one-to-one match.
jj1, jj2, j3	integers	1st point, last point, stride of the second contour. The **pset, get**, setname option described above also applies here.
xcen, ycen, zcen	reals	Relative origin by which the points are scaled.
irz	1	Absolute zoning flag.
	0	Reflective zoning flag.
irratio		Ratio zoning flag. Use that distribution over all point pairs.
	0	Ratio zoning flag off.
	1	Ratio zoning flag on.
	3	Ratio zone on longest line between point pairs, then use that distribution over all point pairs.
rrz	integer	Ratio zoning amount.
2dxy1	2dxy1	Generate points in the same fashion as the **2dxy0** except that there is no one-to-one match.

rzdsab

Zones between surfaces generated by SABRINA.

rzs

Builds a sphere by generating coordinates of points and also modifies zoning by ratio-zoning point distributions.

Format

rzs / itype / nr / npt / xirad, xorad / xcen, ycen, zcen / iz / irat,rz

rzs Parameters

Parameters	Type/Value	Description/Default
itype		Defines what type of sphere will be generated.
	1	Generates a sphere by gridding the faces of a cube and then projecting the vertices onto a sphere. The number of nodes per shell is of the form 6*i**2. Call **connect** to generate connectivity information.
	2	Generates a sphere by subdividing an icosahedron placed on the surface of a sphere. Call **connect** to generate connectivity information.
	diamond	Generates the points for one diamond of the icosahedron. Call **connect** to generate connectivity information. Icosahedralm gridding is made up of 10 diamonds per shell. Each diamond is made up of n**2 nodes (where n must be of the form 2**i+1). There are 2 nodes (the poles of the sphere) at which 5 diamonds meet and 10 nodes where 3 diamonds meet; hence there are a minimum of 12 nodes per shell. The number of nodes per shell can be 12, 42, 162, 642,...
nr	integer	The number of radial shells.
npt	integer	The upper limit of the number of points in a shell.

rzs Parameters

Parameters	Type/Value	Description/Default
xirad, xorad	reals	The inner and outer radii of the sphere. For itype=8 reverse inner and outer radii.
xcen, ycen, zcen	reals	Coordinates of the center of the sphere.
iz	0	mins and maxs are used as cell centers.
	1	then mins and maxs are used as cell vertices.
irat	0	Ratio zoning off.
	1	Ratio zoning on.
rz	real	Ratio zoning value - distance is multiplied by the value for each subsequent point.

Examples

rzs / 8 / 5 / 162 / 1.0, 0.5 / 0., 0., 0. / 1, 0, 0.0
rzs / **2** / 5 / 162 / 0.5, 1.0 / 0., 0., 0. / 1, 0, 0.0
rzs / **diamond** / 5 / 162 / 1 ,.5 / 0, 0, 0 / 1, 0, 0

rzst

Generates points on the surface of spheres or parts of spheres defined by the input parameters. The resultant point distribution should produce similar sized triangular faces. The surfaces are defined with spherical coordinates (r,theta,phi).

Format

rzst / nr, ntheta, nphi / rmin, thetamin, phimin / rmax, thetamax, phimax / xcen, ycen, zcen / irz, itz, ipz / irratio, itratio, ipratio / rrz, rtz, rpz

rzst Parameters

Parameters	Type/Value	Description/Default
add	add	Find the intersection of mesh1 and mesh2. Refine mesh1 where it overlaps mesh2 based on the refine_factor. After the refine steps have been done, the intersection of mesh1 and mesh2 is found, elements that overlap are deleted from mesh1 and mesh2 is appended to mesh1 to create mesh3.
Parameter	Type/Value	Description/Default
nr	integer	The radius of the sphere. Determines the number of spheres.
ntheta	integer	The angle(0 to 180) in degrees from the z-axis. Determines the number of points along theta.
nphi	integer	The angle(0 to 360) in degrees from the x-axis in the xy plane. Determines the number of points along the largest arc with a constant theta.
rmin	integer	Minimum number of spheres.
thetamin	integer	Minimum number of points along theta.
phimin	integer	Minimum number of points along the largest arc with a constant theta, determining part of the surface.
rmax	integer	Maximum number of spheres.
thetamax	integer	Maximum number of points along theta.
phimax	integer	Maximum number of points along the largest arc with a constant theta, determining part of the surface.
xcen, ycen, zcen	reals	Describe the center of the spheres.
irz	0	The reflective zoning flag.
	1	The absolute zoning flag.

rzst Parameters

Parameters	Type/Value	Description/Default
itz	integer	
ipz	integer	
irratio	0	The ratio zoning flag is off.
	1	The ratio zoning flag is on.
itratio		The ratio zoning parameters for theta.
ipratio		The ratio zoning parameter for phi.
rrz		The ratio zoning factor for r.
rtz		The ratio zoning parameters for theta.
rpz		The ratio zoning parameters for phi.

sabangle

Generate a file containing a set of angles that SABRINA will use as its angular distribution for generating points.

Format

sabangle / 1st point, last point, stride / file_name

sabangle Parameters

Parameters	Type/Value	Description/Default
1st point, last point, stride	integers	
file_name	char*32	Default file_name sabangle.

sabplot

Generate a file containing SABRINA body and cell data converted from **surface** and **region** or **mregion** definitions.

Format

sabplot / file_name / region_type

sabplot Parameters

Parameters	Type/Value	Description/Default
file_name	char*32	Default is sabplot.
region_type	region	
	mregion	

sample

Used to turn TALLY sampling on or off and to set the sampling interval (ms).

Format

sample / **on** | **off** / [interval]

sample Parameters

Parameters	Type/Value	Description/Default
on | off	on	Turning on also initializes the sampling, sets the sampling interval, and opens the sample file SMPxxxy, where xxx is the 3-character code identifier (gen, phy, plt, lnk for the generator, physics, plot, link code) and y is the suffix.
	off	Turning off also flushes the sample buffers and closes the sample file. This is the default.
interval		

Examples

sample / on

use 4 ms interval

sample / on / interval
sample / off

scale

Scale a point distribution.

Format

scale / ifirst, ilast, istride **/** scaling_option / geom / iscale, jscale, kscale / xcen, ycen, zcen

scale Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride	integers	Defines the point distribution.
scaling_option	absolute	The scaling factors are constants added on to the existing coordinates. That is, absolute is really a translation rather than a rescale.
	relative	the scaling factors are unitless multipliers with reference to some geometric center (xcen, ycen, zcen).
geom	xyz	Cartesian
	rtz	Cylindrical
	rtp	Spherical
iscale, jscale, kscale	integers	Scale factors. The letters i,j, and k in the scale factors correspond to coordinates specified by one of the geometry types. For example, if geometry = **rtz** then iscale = radial coordinate, jscale = theta coordiante, and kscale = z coordinate. If geometry = **rtp** the iscale = radial coordiante, jscale = theta coordinate and kscale = phi coordinate.
	pset, get name	

scale Parameters

Parameters	Type/Value	Description/Default
xcen, ycen, zcen	reals	Geometric center.

scalec

Scale a set of points to a contour. This command apparently does NOT work properly at all.

Format

scalec / ii1, ii2, ii3 / inoutside / iopt / jji, jj2, jj3 / kk1, kk2, kk3 / igeom / xscale, yscale, zscale / xcen, ycen, zcen

scalec Parameters

Parameters	Type/Value	Description/Default
ii1, ii2, ii3	integers	
inoutside	all	
	inside	
	outside	
iopt	2dxy	
jj1, jj2, jj3, kk1, kk2, kk3	integers	
igeom	rtz	
	rtp	
xscale, yscale, zscale	reals	
xcen, ycen, zcen	reals	

search

This is the main command for generating the connectivity list.

Format

search / isrchopt / ifirst, ilast, istride

search Parameters

Parameters	Type/Value	Description/Default
isrchopt	integer	
ifirst, ilast, istride	integers	

setjtet

Sets **jtet** values at tetrahedron faces formed by sets of specified points to -(mbndry+|jtetset|), provided those faces belong to external boundaries.

Format

setjtet / jtetset / ifirst, ilast

setjtet Parameters

Parameters	Type/Value	Description/Default
jtetset	integer	
ifirst, ilast	integers	

setpts

Sets point types and material regions by calling **surfset** and **regset** routines. Generates constraint table.

Format

setpts
setpts / no_interface
setpts / **closed_surfaces** / **reflect**

setpts Parameters

Parameters	Type/Value	Description/Default
no_interface	no_interface	This allows one to set the **imt** values of nodes without getting any nodes labeled **imt**=interface. This is useful if you do not want **settets** to create parent child chains at interface points. The actual **imt** value given may be determined by roundoff error so should not be used in cases where there are a large number of interface points. It is useful for setting **imt** values of an **rzbrick** mesh in which interface points only occur due to the coincidental point very near the geometry defining surface.
closed_surfaces	closed_surfaces	This option works with geometries in which all **regions** and **mregions** are defined by closed surfaces. The nodes that fall on more than one surface are labeled as interface nodes. Nodes which fall on exactly one surface are labled external reflective boundary nodes. Nodes which are external boundary interface nodes are not labeled correctly and **resetpts / itp** must be called to correct the point types. An input file using the closed_surfaces option is given in the Examples.
reflect	reflect	

Note

The material **imt1** attribute is set based on the **mregion** commands. Interior and external boundary nodes should be assigned to exactly one **mregion**; these nodes will be assigned an **imt1** value that corresponds to the **mregion.** A node which is on an interface will be assigned to any **mregion** and will be given an **imt1** value of **interface** (an integer equal to one more than the number of material regions.) Node types are assigned based on whether a node is on a surface and what type the surface(s) is.
This command must be executed before **search.**

Examples

close_surfaces option

cmo / create / s1 / / / tri
read / avs / surfaces

cmo / create / 3dmesh
surface / surface1 / inticons / sheet / s1
copyts / 3dmesh / s1
cmo / release / s1
cmo / create / 52 / / / tri
read / avs / surf2.avs
surface / surface2 / inticons / sheet / s2
copypts / 3dmesh / s2
cmo / release / s2
region / release / s2
region /r1 / **le** surface1
region / r2 / **le** surface2
mregion / mr1 / **lt** surface1
mregion / mr2 / **lt** surface2
setpts / closed_surfaces / reflect
search
settets
resetpts / itp

settets

Set element color **(**itetclr) and create child points at interior interfaces. This command can also be used to set the node color from the element color.

Format

settets / [type_option]

Note: If there is no type_option, it does the same as **settets / color_tets**

settets Parameters

Parameters	Type/Value	Description/Default
type_option	parents	Has exactly the same behavior as **settets** except that existing values of itetclr are used for elements containing non-interface nodes.
	geometry	Sets the element color based on the material region containing the centroid of the element for all elements; existing values of itetclr are ignored.

settets Parameters

Parameters	Type/Value	Description/Default
	color_tets	Sets the color of all tets using the following tests. If the element contains a non-interface point, the element color is set to this value. If the element is comprised entirely of interface points, the centroid of the element is calculated and the material region containing this point is determined; the element color is set to this material. If the centroid of the element is not in any material region, the centroid must be on an interface surface. In this case all vertices of the element are examined and the material common to all vertices is selected as the element color. Default, if no type_option.
	color_points	Sets the node material (imt1) from the element color (itetclr); existing values of itetclr are not changed.
	newtest	Has the same behavior as **settets** except that existing positive values of itetclr are not changed.
	repair	Has the same behavior as **settets** except that all elements are checked to see if they contain non-interface nodes with different material values (imt1). If so, these nodes are changed to interface nodes and the facets containing them are marked as material interfaces. Intended for use only when multi-material tets exist.
	normal	Assigns the **itetclr** array of a triangle mesh an integer value depending on the normal vector direction. There are 26 possible direction that correspond to the 6 faces, 12 edges and 8 corners of a cube. In general most triangles will be assigned one of 6 values which correspond to the 6 sectors which are within 45 degrees of the 6 unit vectors: 1 0 0 , 0 1 0 , 0 0 1, -1 0 0, 0 -1 0, 0 0 -1 NOTE: Valid only for triangle mesh.

smooth

The **smooth** command smooths 2D or 3D mesh objects. For adaptive smoothing see the **radapt** command. **smooth** takes a 2D or 3D mesh object and moves nodes, without changing the connectivity of the grid, in order to improve the aspect ratios and distribution of elements in the mesh.

Format

smooth / position / esug | mega | geometry | elliptic | random / [ifirst, ilast, istride] / [control]

smooth / position / lpfilter / / [ifirst, ilast, istride] / [filtdeg] / [k_pb] / network

smooth / position / aspect / / [ifirst, ilast, istride / toldamage]

smooth / position / laplace / [ifisrt, ilast, istride] / [rlxwt] / [ntimes] / [nwtty] / [useisn] / [extrnbr]

smooth Parameters

Parameters	Type/Value	Description/Default
position	position	This results in the positions of the nodes being changed.
smooth_algor		There are eight smoothing algorithms available. **esug**, **elliptic** and **random** are for 2D grids, **laplace**, **aspect** and **lpfilte**r for 2D or 3D grids, **mega** and **geometry** are for 3D grids.
	esug	Elliptic Smoothing for Unstructured Grids. This is the default for 2D mesh objects. It can only be used on triangular 2D mesh objects. (Ref.: Ahmed Khamayseh and Andrew Kuprat, "Anisotropic Smoothing and Solution Adaption for Unstructured Grids", Int. J. Num. Meth. Eng., Vol. 39, pp. 3163-3174 (1996).)
	mega	Minimum Error Gradient Adaption. This option creates a smoothed grid which is adapted to the standard function with constant Hessian f(x,y,z)=x2+y2+z2. Can be used on hybrid 3D meshes and guards against mesh folding. Adaption to this function creates a uniform isotropic mesh. The code variable **maxiter_sm** (default=25) controls the maximum number of **mega** iterations. The value of **maxiter_sm** may be changed using the **assign** command (**assign**///**maxiter_sm**/10). (Ref.: Randolph E. Bank and R. Kent Smith, "Mesh Smoothing Using A Posteriori Error Estimates", SIAM J. Num. Anal. tol. 34, Issue 3, pp. 979-997 (1997).)

smooth Parameters

Parameters	Type/Value	Description/Default
	geometry	Geometry ("plump element") adaption. Default for 3D. Can be used on hybrid 3D meshes It uses the **mega** algorithm but retains only the leading geometry term; the term containing the Hessian has been dropped. This algorithm guards against mesh folding
	elliptic	Similar to esug but the 'guards' which prevent a grid from folding are turned off. (Thus esug is preferred.)
	random	a node's position is set to a randomly weighted average position of its neighbors. 'Guards' keep the elements from inverting.
	lpfilter	This smooths surface networks by a low-pass filtering of the coordinate data. (filtdeg default = 30, k_pb default = 0.1)
	aspect	Adjusts node positions such that the aspect ratio of the elements is improved. The default damage tolerance for **smooth//aspect** is infinity, so it can be used as a general smooth which has the effect of improving worst aspect ratio.
laplace	laplace	On a 3D tetahedral mesh moves a node to the average position of its neighbors where neighbor is defined as the set of nodes connected to the candidate node by an edge where the node types (itp1) and node constraints (icr1) are a 'subset' of the candidate node type and constraints. A node will not be moved if the result is an inverted element.
ifirst, ilast, istride	integers	
control	real	An optional control value between zero and one for options **esug**, **mega**, **geometry**, **elliptical** and **random** affects the amount of node movement.
	0.0	Results in the standard smoothing scheme. This is the default.
	> 0.0 < 1.0	Increasing towards 1.0 causes the scheme to be progressively more controlled (moving the mesh less).

smooth Parameters

Parameters	Type/Value	Description/Default
	1.0	There is no mesh movement whatsoever.
	ellipitcal	
	random	
filtdeg		
k_pb		
network	network	**network** smoothing applies to a network of curves in 2D or 3D, or to a network of surfaces in 3D. The materiality of the cells (if any) is ignored.
toldamage		
rlxwt		Wight for underrelaxed Laplacian smoothing. Default is 0.5.
ntimes		Number of smoothing iterations. Default is 5.
nwtty		Number of attempts to not tangle the mesh by halving the smoothing weight. Default is 3.
useisn	1	Interface nodes are smoothed based along a multimaterial edge with all the same materials as the candidate node. Default is 1.
	0	Interface nodes are smoothed based on all interface neighbors.
extrnbr	inclusive	Do not restrict neighbors. Default is **inclusive**.
	exclusive	Restrict neighbors to nodes in pset.

Examples

smooth

Smooth all nodes in the mesh using **esug** in 2D or **geometry** in 3D.

smooth / / / 1,0,0 / 0.5

Smooth all nodes in the mesh, using controlled smoothing with control =0.5

source

Defines source tables for energy, temperature, pressure, or velocity source.

Format

source / source type / source number / mass / multiplier / time shift / subtype / geometry / xcen , ycen , zcen

lm / 0 / 0
time1 , A001 , ... ,
timen , A00n
lm / 1 / 0
time1 , A101 , B101 , ... ,
timen , A10n , B10n
lm / 1 / 1
time1 , A111 , B111 , ... ,
timen , A11n , B11n
. . .
. . .
. . .
lm / lmax / lmax
time1 , Almaxlmax1 , Blmaxlmax1 , ... ,
timen , Almaxlmaxn , Blmaxlmaxn
end

source Parameters

Parameters	Type/Value	Description/Default
source_type	energy	
	temp	
	press	
	vel	

source Parameters

Parameters	Type/Value	Description/Default
source_number		Referenced in a subsequent **zq** command assigning the number to a set of mass points, a mass value, a source multiplier, a time shift, a source subtype, a source geometry (currently only **rtp**), and the location of the origin of the source geometry. For the energy source, the source value in the table is a specific energy rate for subtypes of 0 and 1 and a specific energy for subtypes of 2 and 3. Also for energy subtypes of 1 and 3, the source value is divided by the mass argument. Currently, only spherical harmonics sources are allowed --- S(mu , phi) = A00 + SUM (Plm * (Alm * COS (m * phi) + Blm * SIN(m * phi))) where mu = COS(theta) and Plm is an associated Legendrepolynomial in mu. The SUM is from l = 1 to lmax and m = 0 to l.
mass		
multiplier		
time_shift		
subtype		
geometry		
xcen, ycen, zcen		
lm	lm	Currently, the time values must be the same for all **lm** datasets associated with a given source. For a given source, there must be (lmax + 1) * (lmax + 2) / 2 **lm** data sets.
0 - lmax		
time1 - timen		The constants in the source function are functions of time and are input after the **source** command line.

source Parameters

Parameters	Type/Value	Description/Default
a001 – a00n b101 – b10n		
end	end	

sphbound

Defines a reflective boundary with a (r,theta,phi)-command set.

Format

sphbound / 1 / ibd / ibtp / radius / theta1, phi1 / theta2, phi2 / xcen, ycen, zcen

sphbound Parameters

Parameters	Type/Value	Description/Default
1		
ibd		The reflective boundary number.
ibtp	2	Moving reflective boundary.
	3	Infinite reflective boundary (plane).
	4	Finite reflective boundary (forms a quad in 3D).
radius		
Radius theta1, phi1 theta2, phi2		Define the limits of the boundary. The theta angle (in deg.) is in reference to the x-axis in the xy-plane and the phiI angle (in deg.) is taken with respect to the z-axis.
xcen, ycen, zcen		Center of the spherical boundary.

surf3d

Generate points on a general 3D reference surface using a set of points as shooting rays.

Format

surf3d / interp_pts / map_mat /1st_surf, last_surf, surf_stride / 1st rays, last_rays, rays_stride / xcen, ycen, zcen

surf3d Parameters

Parameters	Type/Value	Description/Default
interp_pts	integer	Number of interpolation points. Default is 4.
map_mat		Maerial to map. Default is 0 (all materials).
1st_surf, last_surf, surf_stride		First 3D surface, last 3D surface, and 3D surface stride.
1st rays, last_rays, rays_stride		First rays, last rays, and rays stride.
xcen, ycen, zcen		X center, y center, and z center.

Note

The points generated are filtered based on the material selected.

surface

Defines a boundary surface or interface.

Format

surface / isurname / ibtype **/ box** / xmin,ymin,zmin / xmax,ymax,zmax
surface / isurname / ibtype/**cone** / x1,y1,z1 / x2,y2,z2 / radius
surface /isurname / ibtype / **cylinder** / x1,y1,z1 / x2,y2,z2 / radius
surface / isurname / ibtype / **ellipse** / x1,y1,z1 / x2,y2,z2 / x3,y3,z3 / ar, br, cr
surface / isurname / ibtype / **parallel** / x1,y1,z1 / x2,y2,z2 / x3,y3,z3 / x4, y4, z4
surface / isurname / ibtype / **plane** / x1,y1,z1 / x2,y2,z2 / x3,y3,z3
surface / isurname / ibtype / **planexyz** / x1,y1,z1 / x2, z2 / x3,y3,z3

surface / isurname / ibtype / **planertz /** radius1,theta1,z1 / radius2, theta2, z2 /
radius,zcen
surface / isurname / ibtype / **planertp** / radius1,theta1,p
surface / isurname / ibtype / **sheet** / cmo_name
surface / isurname / ibtype / **sphere** / x_center, y_center, z_center,radius
surface / isurname / ibtype / **tabular** / x1, y1, z1 / x2, y2, z2 / rz | rt /**&**
r1, z1 **&**
r2, z2 **&**
r3, z3 **&**
....
rn, zn **&**
end
surface / isurname / ibtype / **tabular** / x1, y1, z1 / x2, y2, z2 / rz | rt /**&**
r1, theta1 **&**
r2, theta2 **&**
r3, theta3 **&**
...
rn,thetan **&**
end

surface Parameters

Parameters	Type/Value	Description/Default
isurname	char*32	Name of surface. Must be unique for each surface defined by **surface**.
ibtype		Boundary types
	reflect	External, constrained boundary. Assigned icrl values corresponding to the surfaces on which the nodes sit.
	free	External, unconstrained boundary.
	intrface	Interior, unconstrained material interface.
	intrcons	Interior, constrained material interface. Assigned icrl values corresponding to the surfaces on which the nodes sit.
	virtural	Virtual constrained interface. Assigned icrl values corresponding to the surfaces on which the nodes sit.
box	box	

surface Parameters

Parameters	Type/Value	Description/Default
cone	cone	Where point 1 is the vertex and point 2 is the top center of the cone with radius from that point. A cone is finite but open. To create a closed cone cap the open end with a plane.
cylinder	cylinder	Where point 1 is the bottom center and point 2 is the top center of the cylinder. radius is the radius of the cylinder. Cylinders are open but finite. To create a closed cylinder cap both ends with planes.
ellipse	ellipse	Where point 1 is the center of the ellipsoid and point 2 is on the a semi-axis (new x), point 3 is on the b semi-axis (new y), and ar, br, cr are radii on their respective semi-axes.
parallel	parallel	Where points 1, 2, 3 are the front left, front right and back left points of the base and point 4 is the upper left point of the front face.
plane	plane	
planexyz	planexyz	
planertz	planertz	
planertp	planertp	
sheet	sheet	Sheet surfaces may be input by specifying a cmo_name. The Mesh Object must be either a 2D quad Mesh Object or a 2D triangle Mesh Object. A discussion of inside and outside with respect to sheet surfaces is presented after the EXAMPLES section.
sphere	sphere	

surface Parameters

Parameters	Type/Value	Description/Default
tabular	tabular	Where point 1 and point 2 define the axis of rotation for the tabular profile with point 1 as the origin. This is followed by pairs of profile descriptors depending on the value of geom.Ifgeom is set to **rz**, then the r value is a radius normal to the axis of rotation and z is the distance along the new axis of rotation. If geomis set to **rt** then theta is the angle from the axis of rotation at point 1 andris the distance from point 1 along theta. The first pair must start on a new line and all lines must contain pairs of data. The last pair of data must be followed by end.
xmin,ymin,zmin / xmax,ymax,zmax	reals	
x1,y1,z1 / x2,y2,z2	reals	
radius	reals	
x1,y1,z1 / x2,y2,z2 / x3,y3,z3	reals	
ar,br,cr	reals	
radius1,theta1,z1 / radius2,theta2,z2	reals	
radius,zcen	reals	
radius1,theta1,p	reals	
cmo_name	char*32	
x_center,y_center,z_center, radius	reals	

surface Parameters

Parameters	Type/Value	Description/Default
rz|rt	reals	
r1,z1 **&** r2,z2 **&** r3,z3 **&** .... rn,zn **&** **end**	reals	
r1,theta1 **&** r2,theta2 **&** r3,theta3 **&** ... rn,thetan **&** **end**	reals	

Special Notes

The command **settets** will generate parent/child node chains (isn1) for nodes on **intrface** or **intrcons.**

Surfaces which have different materials on either side of the surface virtual interfaces do not separate material regions but are intended to identify other structural features of a geometry.

Inside/outside with respect to **sheet surfaces** will be determined by the following algorithm:

- For the point being considered, p, find the nearest sheet triangle and the closest point, q, to p that lies on that triangle.
- Construct the vector *d*, from q to p.
- Construct the outward normal to the triangle, *n*. The outward normal is constructed using the right hand rule and the order of the points in the sheet. Sheets may be specified as quad Mesh Object (i.e. a 2 dimensional array of points containing the coordinates of the corners of each quad). Either two triangles (divide each quad in two using point (i,j) and (i+1,j+1)) or four triangles (add a point in the center of the quad) are generated by each quad. Applying the right hand rule to the points (i,j), (i+1,j), (i+1,j+1) gives the direction of the normal for all triangles created from the quad.
- ide. If $d * n > 0$ the point is outside. If $d\ n * n = 0$, and if p is on the triangle then p=q and p in on the triangle.

- * If $d * n = 0$ and p is not on the triangle then p is outside.

One implication of this definition is that the concept of shadows cast by open sheets no longer is valid. Sheets may be considered to extend to the boundary of the geometry.

surfpts

Generates points on boundary surfaces previously defined by the surface or region command. The points are generated by shooting rays through a user specified set of points from an origin point, line or plane and finding the surface intersection of each ray.

Format

surfpts / itype / iname / iregpt / ifirst, ilast, istride /geom. / ray_origin
surfpts / itype / iname / iregpt / **pset get pset_name /** geom. / ray_origin

surfpts Parameters

Parameters	Type/Value	Description/Default
itype	surface	
	region	
iname	char*32	Name of the surface or region.
iregpt	inside outside both	Determines the point location for a region. **Inside, outside** or **both** refers to the surfaces.
ifirst, ilast, istride		Define the set of points to shoot rays through.
geom	xyz	Points 1, 2, 3 define the plane to shoot rays from that are normal to the plane.
	rtz	Points 1, 2, define the line to shoot rays from that are perpendicular to the line.
	rtp	
	points	

surfpts Parameters

Parameters	Type/Value	Description/Default
ray_origin		
pset get pset_name		Define the set of points to shoot rays through.

Specific Geometric Formats

surfpts / itype / iname / iregpt / ifirst, ilast, istride / **xyz** / x1, y1, z1 / x2, y2, z2 / x3, y3, z3

Where points 1, 2, 3 define the plane to shoot rays from that are normal to the plane.

surfpts / itype / iname / iregpt / ifirst, ilast, istride / **rtz** /x1, y1, z1 / x2, y2, z2

Where points 1, 2, define the line to shoot rays from that are perpendicular to the line.

surfpts / itype / iname / iregpt / ifirst, ilast, istride / **rtp** / xcen, ycen, zcen
surfpts / itype / iname / iregpt / ifirst, ilast, istride / **points** / iffirst, iflast, ifstride

tettest

Tettest performs tests on tets.

Formats

tettest / options
tettest / **vol** / nprint, vlimit / ifirst, ilast, istride

tettest Parameters

Parameters	Type/Value	Description/Default
options	all	Performs all the tests.
	jtet	Checks for the consistency of the JTET array.

tettest Parameters

Parameters	Type/Value	Description/Default
	o	Overlap checks for overlapping volumes.
	multimat	Checks for multimaterial tets
	holes	Tests for holes in the mesh.
	faces	Checks for faces that are shared by more than 2 tets.
	connect	Test the connections for consistency.
vol	vol	Looks for tetrahedra in the list that have a volume that is less than the specified vlimit parameter. It also prints the nprint tets with the smallest volumes.
nprint		Tets to be printed in **vol** option.
vlimit		Volume limit for the **vol** option.
Ifirst, ilast, istride		

title

Used to set the problem title (if an argument is specified) or to retrieve the title (if no argument is specified).

Format

title

title / tile_name

title Parameters

Parameters	Type/Value	Description/Default
title_name	char*1000	The title may contain up to 1000 characters, but in a number of applications the code will use only the first 80. Delimit the argument with double quotes to prevent parsing out of blanks, commas, etc. The title is passed between controllees through the restart files.

trans

Translates a set of points.

Format

trans / ifirst, ilast, istride / xold, yold, zold / xnew, ynew, znew

trans / ifirst, ilast, istride / **center** | **zero** | **original** / [**xyz** | **rtp** | **rtz**] [xdim, ydim, zdim]

trans Parameters

Parameters	Type/Value	Description/Default
ifirst, ilast, istride		Set of points to be translated in xyz space.
xold, yold, zold		Old point, translating from.
xnew, ynew, znew		New point to translate to, with a linear translation.
	center	The points set is translated so that 0, 0, 0 is located at the midpoint of min x,y,z and max x,y,z of the mesh.
	zero	The points set is translated so that 0, 0, 0 is located at the min x,y,z of the mesh.
	original	The points set is translated to the original location before **center** or **zero** was called.
	xyz	Currently, only translation.
	rtp	Not available.
	rtz	Not available.
xdim, ydim, zdim		Indicate the axes along which to translate. For example, 1,1,0 or x,y will translate along the x and y axes, the z values will not change.

Example

trans / **pset, get,** mypoints / 0., 0., 0. / 2.0, 2.0, 0.

The points in the **pset** mypoints will be moved 2 in the positive x direction and 2 in the positive y direction.

unit

Checks (changes) the physical units for a give problem. This includes constants, coordinates, etc.

Format

unit / conv / unit type / new unit /old unit
unit / code / unit type / new unit
unit / user / unit type / new unit

unit Parameters

Parameters	Type/Value	Description/Default
options	conv	Conversion units can be used at any time.
	code	Changes the code units. Can only be used at the beginning of a generator input deck. Then all quantities are specified with reference to those consistant set of units.
	user	User units can be used at any time.
unit type		
new_unit		
old unit		

Examples

unit / conv / mass / kg / g
Returns conversion factor to convert from grams to kilograms.

unit / user / mass / kg
Changes the user mass units to kilograms.

unit / code / temp / kev
Changes the code temperature unit to kilo-ev.

variable

Reads and saves a value for a user specified variable name.

Format

variable / variable name / value

variable Parameters

Parameters	Type/Value	Description/Default
variable_name	char*8	User specified variable name.
value		The value is replaced in any subsequent commands that contain the variable name.

view3

Sets viewing angle and distance from the coordinate origin.

Formats

view3 / phi, theta, radius
view3 / xyz / x,y,z

view3 Parameters

Parameters	Type/Value	Description/Default
phi	real	The angle in the xy-plane. Positive values rotate the viewer from looking down the x-axis to looking down the y-axis. Default is 0.0.
theta	real	The tilt upwards toward the Z-axis from the xy-plane. Default is 0.0.
radius	real	Default is 1.0e6.
xyz	xyz	View position with a coordinate.
x	real	Default is 1.0e6.

view3 Parameters

Parameters	Type/Value	Description/Default
y	real	Default is 0.0.
z	real	Default is 0.0.

wedge

Defines the reflective boundaries that makeup a wedge (sides, ends, and the top). This is used to define a 2-D cylindrical geometry. This function uses the **flbound** command to define the reflective boundaries with sets of triplets of points.

Format

wedge / ibd / ibtp / x1, y1, z1 / x2, y2, z2 / x3, y3, z3 / theta

wedge Parameters

Parameters	Type/Value	Description/Default
ibd		The reflective boundary number.
ibtp	2	Moving reflective boundary.
	3	Infinite reflective boundary (plane).
	4	Finite reflective boundary (forms a quad in 3D).
x1, y1, z1 x2, y2, z2		The basic idea is to draw a line from point 1 (x1, y1, z1) to point 2 (x2, y2, z2), this forms the bottom of the wedge.
x3, y3, z3		Point 3 (x3, y3, x3) is used to define the height above the wedge above point 1.
theta		Full angle of spread. Default is 20 degrees.

window

Sets the maximum and minimum plotting window by specifying the lower left and upper right of a cube.

Format

window / xmin, ymin, zmin / xmax, ymax, zmax
window / options
window

window Parameters

Parameters	Type/Value	Description/Default
xmin, ymin, zmin		
xmax, ymax, zmax		
options	save	Saves the current window.
	restore	Restores the saved window.
	resset	Resets the window to the current dump and matset data, but does not fix the window.
	reset	Resets the window to the current dump and matset data, but does not fix the window.
	fix	Fixes the current window. Otherwise the window will change for each dump.
	blank	Prints current window. It is calculated from the current dump.

Note

In plot2d mode, changing the window removes the current zoom levels and creates a new level 1 window.

write

Writes out a region from the generator to a dumpfile.

Formats

write

x3d

Execute one of the other NWGrid controllees from the current controllee.

Format

x3d / **plot | hydro / restart** / file_name / dump_number

x3d Parameters

Parameters	Type/Value	Description/Default
	plot	
	hydro	
restart	**restart**	
file_name	char*32	
dump_numbe r	integer	

z

Prints the status of a given point.

Format

z / mass point num

z Parameters

Parameters	Type/Value	Description/Default
mass point num		

Note

Included in this information is the fitword variables (isq, imt, ipt, ibc, irb, in1, in2, icr, isn), the position (x, y, z) and velocity components (u, v, w), density (rho), radiation temp (trad), radiation press (prad), sound speed (sound), acceleration components (dudt, dvdt, dwdt), the change in density with time (drho/dt) and the change in

energy with time (dedt). Note that the last two values only appear if **z** is used in the generator or physics section of the code.

zq

Set or print node attribute values of a selected set of nodes.

Format

zq / iattribute / ifirst, ilast, istride / value

zq Parameters

Parameters	Type/Value	Description/Default
iattribute	char*32	Valid names can be obtained from the **cmo / status** command.
	vel	Sets velocity vector using spherical coordinates.
	velrtz	Sets velocity vector using cylindrical coordinates.
ifirst, ilast, istride		
value		Attribute value, for all selected nodes, will be set to this value. To print, omit this field.

Note

To print, omit the value field. For printing, attributes are grouped as follows:
Group1: isq,imt,itp (material type and point types)
Group2: x,y,z (coordinates)
To print, specify any one of a group and all will be printed.

Example

zq / **imt** / 1, 100, 2 / material1

will set all odd numbered points between 1 and 100 to material type material 1

zq / xic / 1, 0, 0

will print coordinates of all points

zqread

Reads a file of global arrays data. Currently 5 zone quantities are read with a fixed format. These are

1) Material index.
2) Point type.
3) x-coordinate.
4) y-coordinate.
5) z-coordinate.

Format

zqread / filename / 1st point, last point, stride

zqread Parameters

Parameters	Type/Value	Description/Default
file name	char*32	Default file name is zqread.
1st point, last point, stride		

zqwrite

Writes a file of global arrays data. Currently 5 zone quantities are read with a fixed format. These are

1) Material index.
2) Point type.
3) x-coordinate.
4) y-coordinate.
5) z-coordinate.

Format

zqwrite / file_name / 1st point, last point, stride

zqwrite Parameters

Parameters	Type/Value	Description/Default
file name	char*32	Default is zqwrite.

zqwrite Parameters

Parameters	Type/Value	Description/Default
1st point, last point, stride	char*32	

Documentation

The following ***NWGrid*** documents can be found on the internal, ***NWGrid/NWPhys*** web site. **http://www.emsl.gov/nwgrid**

NWGrid Installation Guide
NWGrid Tutorial
NWGrid Examples
NWGrid Command Reference
NWGrid Data Structures Reference
NWGrid Programmers Reference

Contact

Contact Lynn Trease, llt@pnl.gov, for further assistance and questions.