

# GEN3D

Note: This documentation is very basic at this point. It is here primarily to serve as a listing of the commands in gen3d.

## Abstract:

GEN3D is a three-dimensional mesh generation program. The three-dimensional mesh is generated by mapping a two-dimensional mesh into three-dimensions according to one of four types of transformations: translating, rotating, mapping onto a spherical surface, and mapping onto a cylindrical surface. The generated three-dimensional mesh can then be reoriented by offsetting, reflecting about an axis, and revolving about an axis. GEN3D can be used to mesh geometries that are axisymmetric or planar, but, due to three-dimensional loading or boundary conditions, require a three-dimensional finite element mesh and analysis. More importantly, it can be used to mesh complex three-dimensional geometries composed of several sections when the sections can be defined in terms of transformations of two-dimensional geometries.

SPONSOR: Gregory D. Sjaardema, 1562 (505) 844-2701

KEYWORDS: preprocessor, mesh\_generation

## Contents:

<b>1</b>	<b>Commands .....</b>	<b>3</b>
1.1	Mesh Transformation Commands .....	3
1.2	Mesh Orientation Commands .....	9
1.3	Element Block Type and Boundary Condition Commands .....	11
1.4	Information and Processing Commands .....	12
<b>2</b>	<b>Problems/Bugs .....</b>	<b>13</b>
2.1	Requirements .....	13
<b>3</b>	<b>See Also .....</b>	<b>15</b>

## Documentation:

“GEN3D - A GENESIS Database 2D to 3D Transformation Program” Amy P. Gilkey and Gregory D. Sjaardema, SAND89-0485, March 1989.

“Updates to the mesh generation program GEN3D,” Gregory D. Sjaardema, 1521, dated April 11, 1990.



# 1 Commands

## Mesh Transformation

TRANSLATE	ROTATE	WARP(Point)	WARP(Axis)	SPLINE	PROJECT
SPLINE	PROJECT	TWIST	INTERVAL	SCALE	CHANGE

## Mesh Orientation

REVOLVE	REVCEN	OFFSET	MIRROR	ZERO	SHIFT
---------	--------	--------	--------	------	-------

## Element Block Types

BLOCK	TUNNEL	CENTER	NSETS	SSETS	CHANGE
-------	--------	--------	-------	-------	--------

## Information and Processing

SHOW	LIST	HELP	END	QUIT
------	------	------	-----	------

## Undocumented and experimental

TRANSPLINE

## 1.1 Mesh Transformation Commands

### **TRANSLATE {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, ...**

TRANSLATE causes the 2D mesh to be translated to create the 3D mesh. The number of levels is {ntran}, which is also the number of 3D elements derived from each input 2D element. The total range of the Z coordinate is {tottran} with a gradient of {grad}. The translation is always in the negative Z direction. This command supersedes previous transformation commands. The gradient affects the spacing of the levels. The displacement or thickness of level i is zi where:

$$\begin{aligned} z_1 &= \text{tottran} \times \text{grad} - 1) / (\text{grad}^{\{\text{ntran}\}} - 1) && \text{if grad} \neq 1; \\ &\text{tottran} / \text{ntran} && \text{if grad} = 1 \\ z_i &= z_1 \times \text{grad}^{\{i-1\}} \end{aligned}$$

Multiple translation increments can be specified with a single translate command by repeating the {ntran}, {tottran}, and {grad} parameters on the command line. For example, the following command specifies two translation increments of thickness 1.0 for a total translation of 2.0:

```
TRANSLATE 5 1.0 0.5, 5 1.0 2.0
```

All increments must be specified with a single TRANSLATE command.

### **ROTATE {nrot} <1>, {totdeg} <360.0>, {grad} <1.0>, {cenrot} <0.0>**

ROTATE causes the 2D mesh to be rotated to create the 3D mesh. The number of rotation levels is {nrot}, which is also the number of 3D elements derived from each input 2D element (with the exception of those affected by the CENTER command). The mesh is rotated a total of {nrot} rotations through a total arc of

{totdeg} degrees. The angle of each rotation is equal to {grad} times the previous rotation. The center of rotation {cenrot} and the gradient {grad} are only meaningful if no center element blocks are defined (see the CENTER command).

The gradient affects the rotation of the levels. The angular rotation of level i is thetai where:

$$\text{thetal} = \begin{cases} \text{totdeg} \times (\text{grad} - 1) / (\text{grad}^{\{\text{nrot}\}} - 1) & \text{if grad} \neq 1; \\ \text{totdeg} / \text{nrot} & \text{if grad} = 1 \end{cases}$$

$$\text{thetai} = \text{thetal} \times \text{grad}^{\{i-1\}}$$

Rotation is always counterclockwise. This command supersedes previous transformation commands.

**WARP POINT {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, {radius} <no default>, {edge\_type} <RADIAL>**

WARP POINT causes the 2D mesh to be mapped onto a spherical surface to create the 3D mesh. The spherical surface has a radius of curvature equal to {radius}. The center of curvature is located on the z-axis, and it is a distance of {radius} above the X-Y plane. The number of levels is {ntran}, which is also the number of 3D elements derived from each input 2D element. The total thickness (measured radially) is {tottran} with a gradient of {grad}. Note that {radius} must be greater than the maximum distance from the z-axis to the boundary of the 2D mesh. This command supersedes previous transformation commands. The {edge\_type}, which can be either VERTICAL or RADIAL, determines how the created nodes are generated. If VERTICAL is selected, the x and y coordinates of the generated nodes are equal to the x and y coordinates of the original 2D node. If RADIAL is selected, the x and y coordinates of the generated nodes are calculated to lie on a line from the center of curvature (0.0, 0.0, {radius}) to the coordinates of the warped node (xw, yw, zw) where xw and yw are the coordinates of the original 2D node, and zw is determined such that the distance from the warped node to the center of curvature is equal to {radius}. Figure 2.1 in manual illustrates the VERTICAL edge type, and Figure 2.2 in manual illustrates the RADIAL edge type. The mesh transformation is performed in two parts. First, the warped nodal positions (xw, yw, zw) are calculated by mapping the original 2D mesh onto a spherical surface with a radius of curvature equal to {radius}. The original x and y coordinates of the 2D mesh remain at the same values; the z coordinate is calculated such that the distance to the center of curvature is equal to {radius}.

$$\begin{aligned} xw &= x0 \\ yw &= y0 \\ zw &= \text{radius} - (\text{radius}^2 - x0^2 - y0^2)^{\{1/2\}} \end{aligned}$$

The warped nodal positions are projections parallel to the z-axis onto a spherical surface of radius {radius}; Figure 2.1 in manual illustrates this process. Then, the generated nodal positions are determined by translating either vertically or radially from the warped nodal position. A total of {ntran} translations are

performed through a distance of {tottran} with a gradient of {grad}. Note that the thickness is measured radially for either {edge\_type}. The gradient affects the spacing of the levels. The thickness or length of level i is zi where:

$$z1 = \begin{matrix} \text{tottran} \times \text{grad} - 1 / (\text{grad}^{\{\text{ntran}\}} - 1) & \text{if } \text{grad} \neq 1; \\ \text{tottran} / \text{ntran} & \text{if } \text{grad} = 1 \end{matrix}$$

$$z_i = z1 \times \text{grad}^{\{i-1\}}$$

**WARP {axis} <no default>, {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, {radius} <no default>, {edge\_type} <RADIAL>**

This second form of the WARP command maps the 2D mesh to a cylindrical surface centered on the {axis}-axis to create the 3D mesh. The {axis} parameter must be either X or Y. The cylindrical surface has a radius of curvature equal to {radius}. The center of curvature is located a distance of {radius} above the X-Y plane. The number of levels is {ntran}, which is also the number of 3D elements derived from each input 2D element. The total thickness (measured radially) is {tottran} with a gradient of {grad}. This command supersedes previous transformation commands. The {edge\_type}, which can be either VERTICAL or RADIAL, determines how the created nodes are generated. If VERTICAL is selected, the x and y coordinates of the generated nodes are equal to the x and y coordinates of the projected 2D node. If RADIAL is selected, the x and y coordinates of the generated nodes are calculated to lie on a line from the center of curvature to the coordinates of the warped node (xw, yw, zw) where xw, yw, and zw are the coordinates of the mapped 2D node. The mesh transformation is performed in two parts. First, the warped nodal positions (xw, yw, zw) are calculated by mapping the original 2D mesh onto a cylinder about the {axis}-axis with a radius of curvature equal to {radius}. If {axis} is X, then the original X-coordinate remains at the same value. The generated Y and z coordinates are calculated such that the distance from the generated node to the X-Z plane measured along the cylindrical surface is equal to the X coordinate of the node in the 2D mesh. This is illustrated in Figure 2.2 in manual. If {axis} is Y, the X's and Y's are switched in the above discussion. Then, the generated nodal positions are determined by translating either vertically or radially from the warped nodal position. A total of {ntran} translations are performed through a distance of {tottran} with a gradient of {grad}. Note that the distance is measured radially for either {edge\_type}. The gradient affects the spacing of the levels. The thickness or length of level i is zi where:

$$z1 = \begin{matrix} \text{tottran} \times \text{grad} - 1 / (\text{grad}^{\{\text{ntran}\}} - 1) & \text{if } \text{grad} \neq 1; \\ \text{tottran} / \text{ntran} & \text{if } \text{grad} = 1 \end{matrix}$$

$$z_i = z1 \times \text{grad}^{\{i-1\}}$$

The resulting 3D mesh will have an cylindrical angle of (x{max}/radius) radians if warped about the Y axis, or (y{max}/radius) radians if warped about the X axis, where x{max} and y{max} are the maximum X and Y coordinates in the 2D mesh.

## **SPLINE {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, ...**

where {ntran} is the number of levels, {tottran} is the total transformation distance (thickness), and {grad} is the gradient which affects the spacing of the levels (see page 21 in Reference GEN3D for an explanation of the gradient). Multiple transformation increments can be specified with a single SPLINE command by repeating the {ntran}, {tottran}, {grad} parameters on a single line. Note that the actual thickness of the generated mesh is determined by the input front and back spline surfaces; therefore, the distances entered as {tottran} are the proportional distance of the segments. For example, if the following command was entered:

```
SPLINE 2 1.0 0.5    4 2.0 1.0    2 1.0 2.0
```

then, segments 1 and 3 would each be 25% of the total thickness, and segment 2 would be 50% of the total thickness. Following the SPLINE command line, GEN3D enters the spline input mode in which the various spline options described below can be entered.

### **LINEAR**

the spline data are input as Radius-Z data pairs, and the slopes at the end of the curves are linear slopes.

### **ANGULAR**

the spline data are input as Theta(degrees)-Distance data pairs, where Theta is the angle of the line between the origin ( $X = Y = Z = 0$ ) and the defined point and the Distance is the length of this curve. The slopes at the end of the curves are relative to this curve.

### **FRONT**

the curve data and slope specifications up to the next BACK, END, or EXIT command will the FRONT spline. The front surface Z values are greater (more positive) than the back surface Z values.

### **BACK**

the curve data and slope specifications up to the next FRONT, END, or EXIT command will the BACK spline. The front surface Z values are greater (more positive) than the back surface Z values.

### **LEFT {slope}**

the parameter {slope} specifies the slope of the spline curve at the LEFT end of the curve. The slope is measured in the same units specified in the ANGULAR or LINEAR command. If the slope is not specified, the end conditions of the curve will be set such that the second derivative is equal to zero which is the so-called natural cubic spline.

## **RIGHT {slope}**

the parameter {slope} specifies the slope of the spline curve at the RIGHT end of the curve. The slope is measured in the same units specified in the ANGULAR or LINEAR command. If the slope is not specified, the end conditions of the curve will be set such that the second derivative is equal to zero which is the so-called em natural cubic spline.

## **EXIT or END**

terminate spline input mode and return to general GEN3D command processing.

See Also

Memo: "Updates to the mesh generation program GEN3D,"

Gregory D. Sjaardema, 1521, dated April 11, 1990.

See the author for copies.

## **PROJECT {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, ...**

where {ntran} is the number of levels, {tottran} is the total transformation distance (thickness), and {grad} is the gradient which affects the spacing of the levels (see page 21 in Reference GEN3D for an explanation of the gradient). Multiple transformation increments can be specified with a single PROJECT command by repeating the {ntran}, {tottran}, {grad} parameters on a single line.

## **NORMAL {x\_normal} <0.0>, {y\_normal} <0.0>, {z\_normal} <0.0>**

## **PLANE {x\_normal} <0.0>, {y\_normal} <0.0>, {z\_normal} <0.0>**

NORMAL or PLANE the normal vector to the back surface. The {x\_normal}, {y\_normal}, and {z\_normal} parameters are vector components of the normal vector. The front surface will be projected onto the plane with the specified normal vector. Because of the way GEN3D generates the ddd mesh, the {z\_normal} component of the vector must be negative; if a positive value of {z\_normal} is entered, all of the components will be multiplied by negative one.<sup>footnote</sup>The test version of GEN3D did not do this reversal correctly; instead it always made {z\_normal} negative which made it confusing to determine the surface orientation. This bug has been fixed; however, old input files will now give an incorrect orientation. To use your old input files, either correct the normal vector, or put the keyword DOOLDWAY after the {z\_normal} component. NORMAL or PLANE supersede previous WARP commands.

## **WARP {distance} {CONVEX|CONCAVE}**

WARP projects the front surface (original dd mesh) onto a spherical surface with a radius of {distance}. If CONVEX is specified, the generated mesh will have a spherical or bulbous surface; if CONCAVE is specified, the generated mesh will have a dimpled surface. The distance {tottran} specified in the PROJECT

command is measured to the center of the spherical surface. WARP supersedes previous NORMAL or PLANE commands. Note that the SPLINE transformation can be used if a non-spherical surface is required.

**SCALE {x\_scale} <1.0>, {y\_scale} <1.0>**

SCALE multiplies the X and Y coordinates of the projected surface by the respective scale factors. See the SCLCEN command for the equations used to transform the coordinates.

**SCLCEN {x\_cen} <0.0>, {y\_cen} <0.0>**

SCLCEN specifies the origin of the coordinate system for scaling. This is best illustrated by examining the equations used in the transformation:

$$x_{\text{new}} = (x_{\text{old}} - x_{\text{cen}}) \times x_{\text{scale}} + x_{\text{cen}}$$

$$y_{\text{new}} = (y_{\text{old}} - y_{\text{cen}}) \times y_{\text{scale}} + y_{\text{cen}}$$

**OFFSET {x\_offset} <0.0>, {y\_offset} <0.0>**

offsets the nodal coordinates of the projected surface by the specified {x\_offset} and {y\_offset} which shifts the back surface with respect to the front surface. The offsets are performed after the surface has been projected onto the plane or spherical surface.

**RESET**

resets all parameters to their default values.

**EXIT or END**

terminates the **PROJECT** input mode and returns to normal GEN3D command processing.

See Also

Memo: "Updates to the mesh generation program GEN3D,"

Gregory D. Sjaardema, 1521, dated April 11, 1990.

See the author for copies.

**TWIST {twangl} <0.0>, {twxcen} <0.0>, {twycen} <0.0>, {TRANSLATE}, {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, ...**

**TWIST {twangl} <0.0>, {twxcen} <0.0>, {twycen} <0.0>, {Rotate}, {cenrot} <0.0>, {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, ...**

where {twangl} is the rotational offset in degrees of the front surface with respect to the back surface, {twxcen} and {twycen} specify the center of the twist

rotation, {ntran} is the number of levels, {tottran} is the total transformation distance (distance for the TRANSLATE option, or degrees for the ROTATE option), and {grad} is the gradient which affects the spacing of the levels (see page 21 in Reference GEN3D for an explanation of the gradient). Multiple transformation increments can be specified by repeating the {ntran}, {tottran}, {grad} parameters on a single line. The twist angle per level is determined by dividing {twangl} by the total number of translation increments {ntran}, not by the total translation/rotation distance {tottran}. Note that this means that the incremental twist per level depends on the gradient.

**INTERVAL {ntran} <1>, {tottran} <1.0>, {grad} <1.0>, ...**

where {ntran} is the number of levels, {tottran} is the total transformation distance in units applicable to the currently active transformation option, and {grad} is the gradient which affects the spacing of the levels (see page 21 in Reference GEN3D for an explanation of the gradient). Multiple transformation increments can be specified with a single INTERVAL command by repeating the {ntran}, {tottran}, {grad} parameters on a single line.

**SCALE {axis\_1}, {scale\_1}, {...}**  
**SCALE ALL {scale}**  
**SCALE RESET**

where {axis\_i} is either X, Y, or Z; and {scale\_i} is the factor by which the specified axis coordinates are multiplied. If ALL is specified, all three coordinates are multiplied by {scale}. Scaling can be reset by the command SCALE RESET.

**CHANGE {Material|Nodeset|Sideset} {old\_id} {new\_id}**

where {old\_id} is an existing ID in the original mesh, and {new\_id} is an ID that does not exist in the original mesh. Note that you cannot combine IDs using this command; the ID {new\_id} must not exist in the original mesh, and it must not match an ID created using the NODESET|SIDESET FRONT|BACK commands or the TUNNEL commands.

## 1.2 Mesh Orientation Commands

**REVOLVE {axis\_{1}}, {ndeg\_{1}}, {axis\_{2}}, {ndeg\_{2}}, ... <last selection>**  
**REVOLVE RESET <initial condition>**

REVOLVE causes the transformed 3D mesh to be rotated. Each ({axis}, {ndeg}) parameter pair specifies an axis (X or Y or Z) and the number of degrees to rotate. The axis refers to the “viewing” axis, not to the object axis. The rotations are according to right-hand rule. The center of the rotation is specified by the REVCEN command. Revolutions are cumulative; however, only one center of revolution may be specified. The REVOLVE RESET command resets to no rotation.

**REVCEN {xcen} <2D minimum X coordinate>,  
{ycen} <2D minimum Y coordinate>, {zcen} <0.0>**

REVCEN sets the center of revolution for the REVOLVE command to the point ({xcen},{ycen},{zcen}).

**SHIFT [ADD] {axis\_{1}}, {offset\_{1}}, {axis\_{2}}, {offset\_{2}}, ...  
SHIFT ALL {offset} <0.0>  
SHIFT RESET <initial condition>  
SHIFT {xoff} <0.0>, {yoff} <0.0>, {zoff} <0.0>**

(NOTE: SHIFT is a synonym for OFFSET) SHIFT specifies offsets to be added to the coordinates. If a REVOLVE command has been issued, the mesh is rotated before it is offset. The last form of the offset command is included to maintain compatibility with the offset command in GEN3D. SHIFT ALL offsets all of the coordinates by the specified offset, and SHIFT RESET resets the offsets to zero. Offsets are not cumulative unless ADD is specified, that is, if SHIFT X 0.5 X 1.0 is entered, the X coordinates will be offset by 1.0. if SHIFT ADD X 0.5 X 1.0 is entered, the X coordinates will be offset by 1.5.

**OFFSET [ADD] {axis\_{1}}, {offset\_{1}}, {axis\_{2}}, {offset\_{2}}, ...  
OFFSET ALL {offset} <0.0>  
OFFSET RESET <initial condition>  
OFFSET {xoff} <0.0>, {yoff} <0.0>, {zoff} <0.0>**

(NOTE: OFFSET is a synonym for SHIFT) OFFSET specifies offsets to be added to the coordinates. If a REVOLVE command has been issued, the mesh is rotated before it is offset. The last form of the offset command is included to maintain compatibility with the offset command in GEN3D. OFFSET ALL offsets all of the coordinates by the specified offset, and OFFSET RESET resets the offsets to zero. Offsets are not cumulative unless ADD is specified, that is, if OFFSET X 0.5 X 1.0 is entered, the X coordinates will be offset by 1.0. if OFFSET ADD X 0.5 X 1.0 is entered, the X coordinates will be offset by 1.5.

**MIRROR {axis\_{1}}, {axis\_{2}}, ... <no default>  
MIRROR RESET <no reflections>**

MIRROR causes the transformed 3D mesh to be reflected about a coordinate plane. Each {axis} parameter specifies an axis (X or Y or Z) which is the normal to the reflection plane. Reflections are performed after the mesh has been repositioned by the REVOLVE and OFFSET commands. The MIRROR RESET command resets to no reflection. Reflections are not cumulative, that is, if MIRROR X Y X is entered, only one reflection about the X axis will be performed. If an odd number of reflections are performed, the element connectivity and the sideset face numberings will be correctly reordered.

**ZERO {axis\_1}, {min\_1}, {axis\_2}, {min\_2}, ...**

## **ZERO RESET <no automatic zeroing>**

ZERO sets all {axis\_i} coordinates with an absolute value less than {min\_i} equal to zero. The ZERO RESET command resets to no automatic zeroing. This command is used to zero nodal coordinates that should be equal to zero, but due to roundoff errors they have slightly nonzero values.

## **1.3 Element Block Type and Boundary Condition Commands**

### **BLOCK {block\_id\_{1}}, {block\_id\_{2}}, ... <all element blocks>**

BLOCK defines the specified element blocks as normal blocks. This command supersedes any previous TUNNEL or CENTER commands.

### **TUNNEL {block\_id} <no default>, {start} <1>, {end} <number of levels>, {inc} <1>**

TUNNEL defines the specified element block as a tunnel block. A TRANSLATE command must be in effect before this command is issued. If a ROTATE command is issued, all tunnel blocks are changed to normal blocks. For each tunnel block, a separate 3D element block is created starting at level {start}, with each block having {inc} levels. Any levels after level {end} are put in a single block. For example, the commands

```
TRANSLATE 15, 15.0
TUNNEL 999, 5, 9, 2
```

create five blocks consisting of the following 3D elements (derived from the 2D elements of element block 999):

- 1) the elements in levels 1, 2, 3, and 4,
- 2) the elements in levels 5 and 6,
- 3) the elements in levels 7 and 8,
- 4) the elements in level 9,
- 5) the elements in levels 10, 11, 12, 13, 14, and 15.

The block identifier of the first block is always {block\_id}. The new blocks are assigned consecutive identifiers greater than the maximum existing (and new) identifier.

### **CENTER {block\_id\_{1}}, {block\_id\_{2}}, ... <all element blocks>**

CENTER defines the specified element blocks as center blocks. A ROTATE command must be in effect before this command is issued. The mesh must be rotated a complete quadrant (90, 180, 270 or 360 degrees) and the number of rotation levels must be a multiple of 2 for each 90 degrees of rotation. If em nrot is the number of rotations, there must be at least em nrot/2 elements along the X axis in the center block. If a TRANSLATE command is issued, all center blocks are changed to normal blocks. If center blocks are defined, the center of rotation defined by the ROTATE command is ignored. The center of rotation is the

minimum coordinate of all elements in the center blocks.

**NSETS FRONT or BACK <no default>, {set\_id\_{1}}, {set\_id\_{2}}, ...  
<no default>**

NSETS defines front or back node sets with the given identifiers. The identifiers must be unique from existing node set identifiers and previously defined front and back node set identifiers. Back sets cannot be defined on a 360-degree rotation.

**SSETS FRONT or BACK <no default>, {set\_id\_{1}}, {set\_id\_{2}}, ...  
<no default>**

SSETS is equivalent to the NSETS command except that it defines side sets.

## 1.4 Information and Processing Commands

**SHOW {command} <no parameter>**

SHOW displays the settings of parameters relevant to the {command}. For example, the command SHOW BLOCK displays information about all the element blocks.

**LIST VARS**

LIST VARS displays a summary of the input database. The summary includes the database title; the number of nodes, elements, and element blocks; and the number of node sets and side sets.

**HELP {command} <no parameter>**

HELP displays information about the program command given as the parameter. If no parameter is given, all the command verbs are displayed. This command is system-dependent and may not be available on some systems.

**END**

END ends the command input and starts the database transformation.

**QUIT**

ends the command input and exits the program immediately without writing an output database.

## 2 Problems/Bugs

CENTER BLOCKS -- Incorrect mesh will be generated if center blocks are not contiguous (ie. contain slidelines). Problem will appear at junction of pieces connected by slideline. The nodes on the of the piece will not be generated correctly.

FIX: Generate each piece separately and join together with GJOIN  
Problem involves changing several routines and has only been reported once in two years. Sorry, Greg Sjaardema

### 2.1 Requirements

GEN3D Requires the following Utilities/Files:

- o suplib      Library of common routines for reading/writing  
                 EXODUS files and other support routines.
- o supes      Memory management and Free-field input routines



### 3 See Also

- FASTQ: Generate two-dimensional GENESIS mesh file
- GJOIN: Merge two or more mesh files into single file
- GREPOS: Reposition/Modify mesh file
- NUMBERS: Calculate several parameters of a mesh file
- APREPRO: Algebraic preprocessor for input files.
- BLOT: Plot mesh files.