Sandia National Laboratories
Albuquerque, New Mexico 87185

date: April 11, 1990

to: Distribution

from: Gregory D. Sjaardema, 1521

subject: Updates to the mesh generation program *GEN3D*

Introduction

This memo describes the changes that have been made to the *GEN3D* program since the manual (Reference [1]) was published. The changes include:

- New mesh generation options:
  - SPLINE transformation
  - PROJECT transformation
  - TWIST transformation
  - INTERVAL specification
  - ROTCEN option
- New mesh orientation option
  - SCALE command
- New mesh modification options
  - CHANGE MATERIAL
  - CHANGE SIDESET
  - CHANGE NODESET
- Miscellaneous changes
  - NODESETS = NSETS
  - SIDESETS = SSETS
  - EXIT = END
  - LIST COMMANDS

SPLINE Transformation

A SPLINE transformation option has been added to *GEN3D*. This transformation maps a two-dimensional mesh onto two surfaces of revolution. The surfaces of revolution are generated by sweeping two user-input cubic splines about the Z axis (the original two-dimensional mesh is assumed to lie in the X–Y plane). The three-dimensional mesh is then generated by interpolating the specified number of intervals (*ntran*) between these two surfaces of revolution. The command specification is given below and a step-by-step description of the mesh generation process is given in Appendix B. An example three-dimensional mesh is shown in Figure 1.

SPLINE *ntran* <1>, *tottran* <1.0>, *grad* <1.0>, ...

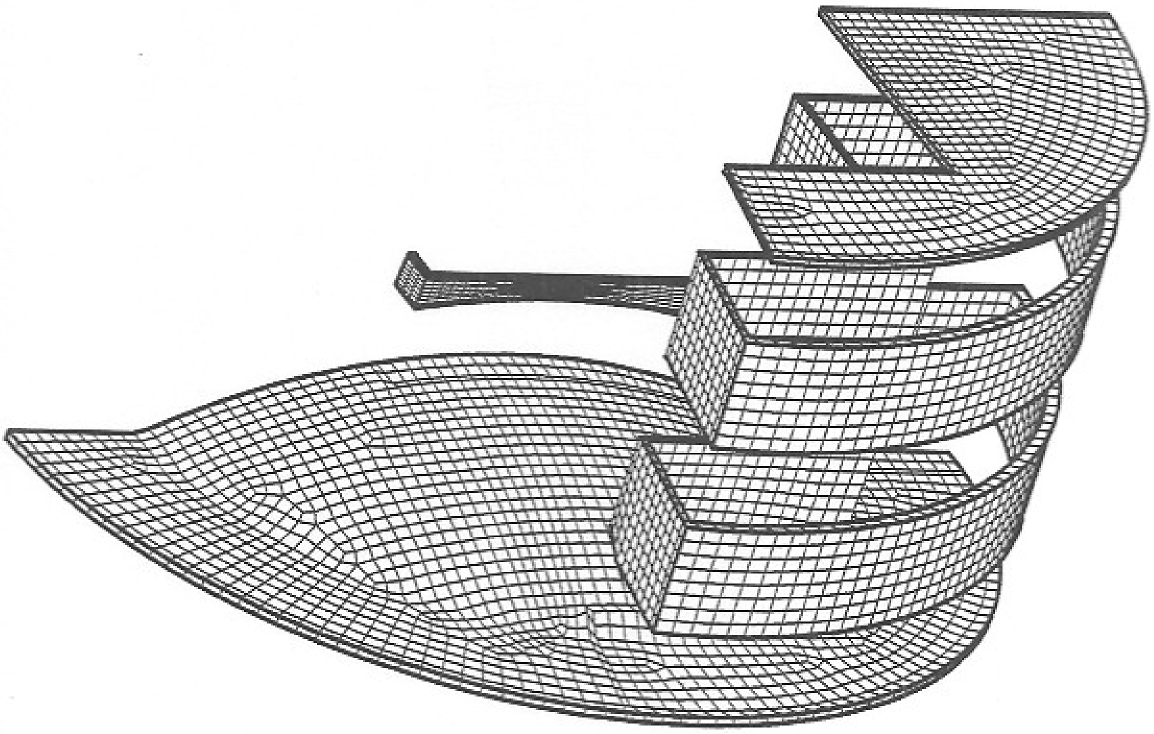

Figure 1. Example Three-Dimensional Spline Mesh. Each piece of this mesh was generated using the SPLINE option.

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 [1] 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. Figures B.2 and B.3 contain example input files which illustrate the use of these commands.

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 define 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 define 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 *natural cubic spline.*

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

Limitations:

1. *GEN3D* does not check the validity of the input front and back spline curves. That is, *GEN3D* will generate meshes with negative volume elements if the front and back splines are reversed or crossed.
2. Currently only surfaces defined by sweeping the curves about the Z axis can be generated. A planned enhancement is to allow for planar surfaces.

```
Project 5 1.0 0.8 5 1.0 1.25
   Normal 0.5 0.5 0.5
   Exit
Exit
```

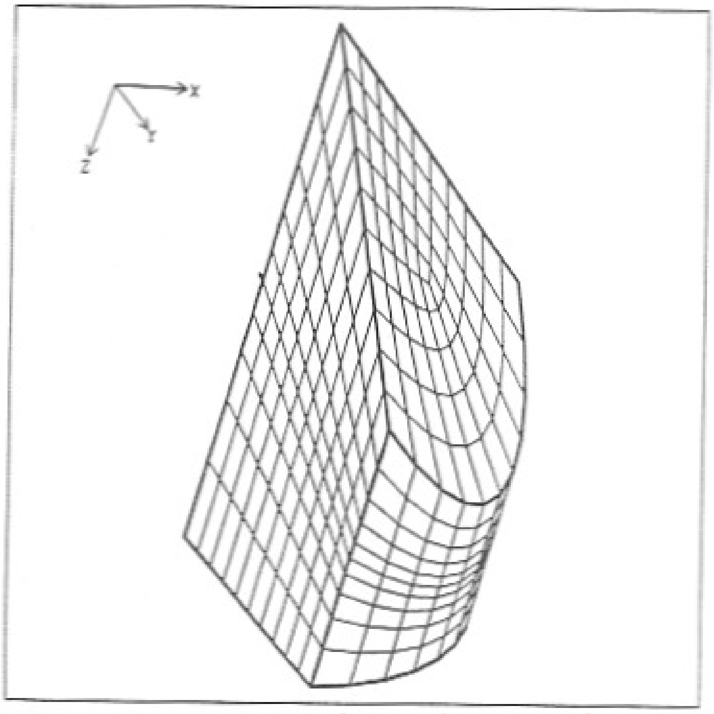

Figure 2. Result of PROJECT transformation: quarter-circle two-dimensional mesh transformed into three-dimensional mesh.

PROJECT Transformation

The PROJECT[1] option generates the mesh by creating elements between the front surface which is the input two-dimensional mesh, and the back surface which is the input two-dimensional mesh projected onto a surface below (in the negative Z direction) the input mesh. The positioning of the back surface is specified in the 'project input mode' which is described below. *GEN3D* enters the 'project input mode' following the PROJECT command line.

All of the options affect only the positioning of the back surface; the front surface is the original two-dimensional mesh. Figure 2 shows an example input file which illustrates the use of these commands, and the generated mesh.

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 [1] 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>

[1] Pronounced with a long O sound

NORMAL or PLANE define 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 three-dimensional 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.[2]

NORMAL or PLANE supersede previous WARP commands.

WARP *distance* CONVEX|CONCAVE

WARP projects the front surface (original two-dimensional 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_center* <0.0>, *y_center* <0.0>

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

$$\begin{aligned} x_{\text{new}} &= (x_{\text{old}} - x_{\text{cen}}) \times x_{\text{scale}} + x_{\text{cen}} \\ y_{\text{new}} &= (y_{\text{old}} - y_{\text{cen}}) \times y_{\text{scale}} + y_{\text{cen}} \end{aligned}$$

OFFSET *x_offset* <0.0>, *y_offset* <0.0>

OFFSET 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

RESET resets all parameters to their default values.

[2] 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.

```
Twist  180.0  0.0  0.0  Translate  18 18.0
Exit
```

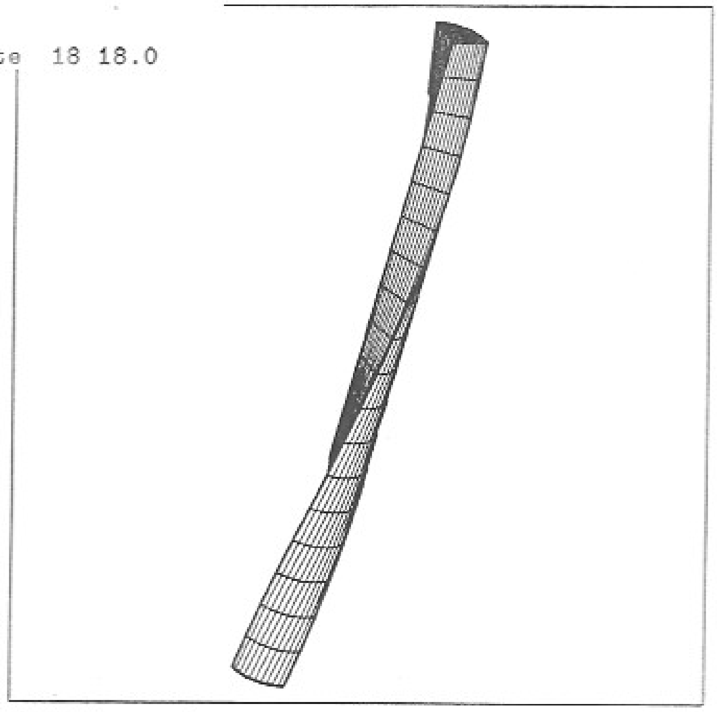

Figure 3. Result of TWIST TRANSLATE transformation: two-dimensional quarter-circle mesh transformed into three-dimensional mesh.

EXIT
END

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

TWIST Command

The TWIST command generates the three-dimensional mesh by twisting the two-dimensional mesh as it is generated. Two options are available: TRANSLATE or ROTATE. The TRANSLATE option generates a mesh similar to the normal *GEN3D* TRANSLATE transformation except that the two-dimensional mesh is twisted as it is translated. Similarly, the ROTATE option generates a mesh similar to the normal *GEN3D* ROTATE transformation except that the two-dimensional mesh is twisted as it is rotated. Figures 3 and 4 illustrate the TWIST/TRANSLATE and TWIST/ROTATE transformations.

The command syntax is:

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 *twxcen* specify the center of the twist rotation,

```
Twist 360.0 0.0 0.0 Rotate -5.0 72 360
Exit
```

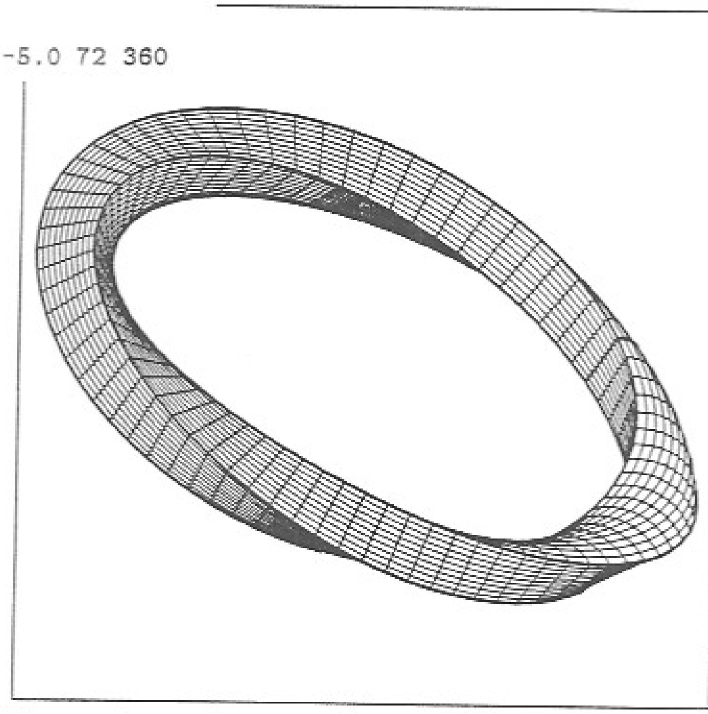

Figure 4. Result of TWIST ROTATE transformation: two-dimensional quarter-circle mesh transformed into three-dimensional mesh.

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 [1] 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.

Limitations:

- If *tottran* is equal to 360 degrees for the TWIST/ROTATE option, the back surface will not be joined to the front surface if they touch. If you need a continuous twisted ring, you will have to specify front and back sidesets and use a tied contact surface in the analysis code.

INTERVAL Command

The INTERVAL command is used to modify the number of intervals, the transformation distance, and the transformation gradient for each of the mesh transformation commands. This command is provided to allow the use of multiple transformation increments in the ROTATE and WARP mesh transformations. For example, a mesh rotated 180 degrees with elements increasing in size from 0 to 90 degrees and decreasing in size from 90 to 180 degrees can be generated by the command:

```
ROTATE
INTERVAL      10 90.0 1.25      10 90.0 0.8
```

where the ROTATE command sets the transformation type and the center of rotation (see page 21 in Reference [1]), and the INTERVAL specifies two transformation increments. The first increment is 10 rotation levels for a total of 90.0 degrees with a gradient of 1.25, and the second increment is 10 rotation levels for a total of 90.0 degrees with a gradient of 0.8. The generated mesh will have 20 rotation levels for a total of 180.0 degrees. The command syntax is:

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 [1] 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.

ROTCEN Command

The ROTCEN command is used to set the center of rotation for the ROTATE command. It is also used with the INTERVAL command to specify a multiple segment rotation transformation. For example, the commands

```
ROTCEN 1.0
INTERVAL      10 90.0 1.25
```

are equivalent to the command

```
ROTATE      10 90.0 1.25 1.0
```

The command syntax is: **ROTCEN** *cenrot* <0.0>

SCALE Command

The SCALE command is used to multiply the coordinates in each of the three coordinate directions by a specified scale factor. The command syntax is:

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 Command

The IDs associated with material blocks, sidesets, and nodesets can now be changed within *GEN3D* using the CHANGE command. The command syntax is:

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.

Miscellaneous Changes and Additions:

A few minor changes have been made to the *GEN3D* command syntax to reduce the chance for errors during code input. The commands NODESETS and NSETS can be used interchangeably to refer to node sets, and similarly the commands SIDESETS or SSETS can be used to refer to side sets. *GEN3D* now recognizes EXIT or END as directives to end command input and begin processing. The command LIST COMMANDS has been added to display all commands. It only shows the command name, not the command syntax.

Future Development Plans

- Enhance the SPLINE transformation to include sweeping the spline curve parallel to the X or Y axis in addition to the current method of sweeping about the Z axis.
- An OFFSET SPLINE option has also been requested. This option would be added to the PROJECT options and would allow the user to specify the X and Y offsets to be applied to a mesh as a function of the Z coordinate. This is needed to generate a three-dimensional mesh of a curvilinear borehole for a penetration analysis.
- Several users have asked that the translation distance in the TRANSLATE command be allowed to be negative (see page 21 in Reference [1]). This will be implemented for most of the transformation options.
- The DELETE Material|Nodeset|Sideset options recently added to *GREPOS* [3] will be added to *GEN3D*.

If you are interested in any of these commands, or need a transformation option not yet implemented, see the author. Additional transformation options can usually be implemented in a very short time especially if they are not required to be completely general.

QA and Information Fields

GEN3D has been modified to comply with the current requirements in the draft Department 1520 Quality Assurance (QA) document [4]. Specifically, *GEN3D* adds its code identification QA field to the Exodus [5] QA records, and it adds an information field identifying the file name of the two-dimensional mesh which is transformed into the output three-dimensional mesh.

The version of *GEN3D* described in this memo is `X1.01.mm`[3].

References

[1] A. P. Gilkey and G. D. Sjaardema, "GEN3D: A GENESIS Database 2D to 3D Transformation Program," SAND89-0485, Sandia National Laboratories, Albuquerque, NM, March 1989.

[2] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes, the Art of Scientific Computing.* Cambridge: Cambridge University Press, 1986.

[3] G. D. Sjaardema, "GREPOS: A GENESIS Database Repositioning Program," SAND90-0566, Sandia National Laboratories, Albuquerque, NM. In preparation.

[4] C. R. Adams, 1522, S. W. Attaway, 1521, T. D. Blacker, 1523, and G. D. Sjaardema, 1521, "Department 1520 software QA committee report," (Memo to Distribution), Sandia National Laboratories, Albuquerque, NM, February 16, 1990.

[5] W. C. Mills-Curran, A. P. Gilkey, and D. P. Flanagan, "EXODUS: A Finite Element File Format for Pre- and Post-processing," SAND87–2977, Sandia National Laboratories, Albuquerque, NM, September 1988.

[3]Department 1520 is in the process of implementing a new QA procedure for all of their codes. A key element of this procedure is the version numbering of the codes. Each code version is identified by an eight character field consisting of four parts. The QA field is in the form: `Qv.MM.mm`. The `Q` field is one of the letters `A` for codes satisfying all QA requirements, `C` for archived codes, and `X` for experimental or development codes. The `v` field is incremented when there are major modifications to the code which require the issuance of a new SAND report. The `MM` field is incremented for major modifications to the code or when new capabilities have been added. The `mm` field is incremented for minor changes and bug fixes. More information about the Department 1520 QA procedure can be found in Reference [4]

Appendix A. Command Summary

SPLINE *ntran, tottran, grad, ...* (Page 1)
    LINEAR
    ANGULAR
    FRONT
    BACK
    LEFT *slope*
    RIGHT *slope*
    END, EXIT

PROJECT *ntran* <1>, *tottran* <1.0>, *grad* <1.0>, ... (Page 4)
    SCALE *x_scale y_scale*
    SCLCEN *x_center y_center*
    OFFSET *x_offset y_offset*
    NORMAL *x_normal y_normal z_normal*
    PLANE *x_normal y_normal z_normal*
    WARP *distance* CONVEX|CONCAVE
    RESET
    EXIT, END

TWIST *twangl, twxcen, twycen,* TRANSLATE, *ntran, tottran, grad, ...* (Page 6)
TWIST *twangl, twxcen, twycen,* ROTATE, *rotcen, ntran, tottran, grad, ...*
INTERVAL *ntran, tottran, grad, ...* (Page 7)
ROTCEN *cenrot* (Page 8)
SCALE $axis_1$, $scale_1$, ... (Page 8)
SCALE ALL *scale*
SCALE RESET
CHANGE {Material|Nodeset|Sideset} *old_id new_id* (Page 9)

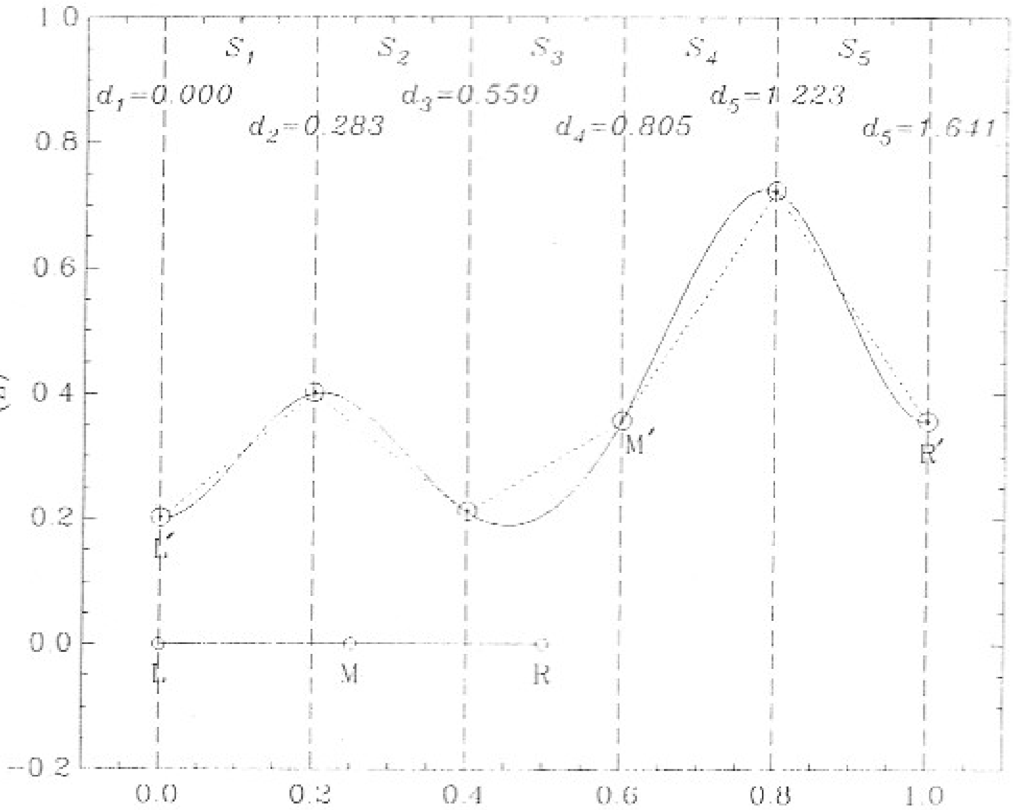

Figure B.1. Example Spline Curve for Spline Mesh Generation Description

Appendix B. Spline Mesh Generation Process:

The following procedure is used to generate a three-dimensional mesh from a two-dimensional mesh using the SPLINE option. Figure B.1 shows an example spline curve which will be used to illustrate the steps in the procedure. The example spline is defined by 6 data points which break the curve into 5 segments S_1 through S_5. Zero slopes are specified at both ends of the curve. Only a single curve is shown in the figure to reduce the clutter; for an actual mesh, the back spline curve would also be needed. The line at Z = 0 represents a radial section through a two-dimensional mesh. The letters L, M, and R represent nodes at the left end, middle, and right end of the mesh, respectively.

1. Generate the cubic spline parameters to fit the data entered by the user using the `SPLINE` and `SPLINT` routines in Reference [2]. The solid line in Figure B.1 is the generated spline curve.
2. Calculate the cumulative chord lengths d_i of each segment of the front and back splines. The chord length of each segment is calculated as the straight line distance between each of the entered data points. This is a good approximation for smooth curves, but it underestimates the distance for non-smooth curves. If the spline does not begin at the Z axis, then the minimum distance of the spline is equal to the distance from the Z axis. The cumulative chord lengths of the example spline are denoted by d_i and shown as dotted lines in the figure. Note that the minimum

distance is equal to d_1, and the maximum distance is equal to d_{NS} where NS is the number of points defining the curve.

3. Determine the minimum and maximum radii $m_{\min}$ and $m_{\max}$ of the input two-dimensional mesh. The radii are measured from the point X = Y = 0.

$$m_{\max} = \max_{i=1,N} \sqrt{x_i^2 + y_i^2} \tag{B.1}$$

$$m_{\min} = \min_{i=1,N} \sqrt{x_i^2 + y_i^2} \tag{B.2}$$

   where, N is the number of nodes in the two-dimensional mesh, and x_i and y_i are the coordinates of node i.

4. The two-dimensional mesh is mapped onto the spline surface of revolution such that the maximum mesh radius ($m_{\max}$) is mapped to the outside edge of the spline surface, and the minimum mesh radius ($m_{\min}$) is mapped to the inside edge of the spline surface. This mapping defines a coordinate transformation where the two-dimensional mesh is uniformly stretched or shrunk onto the spline surface. Note that the distance is measured along the surface. If d_i is the cumulative chord length at point i of the curve defining the spline surface of revolution, and $m_{\min}$ and $m_{\max}$ are the minimum and maximum radii of the two-dimensional mesh, then radial distances l in the two-dimensional mesh will be mapped to distances l' along the spline surface, where

$$l' = d_1 + \frac{(d_{NS} - d_1)}{(m_{\max} - m_{\min})} \times (l - m_{\min}) \tag{B.3}$$

   For example, the point M in the example figure is in the middle of the two-dimensional line. It would be mapped to point M' on the spline curve.

5. Each node in the two-dimensional mesh is mapped to the three-dimensional mesh by the following steps (5a through 5g)

   (a) Determine the polar position (r, θ) of the node in the X–Y plane:

$$r = \sqrt{x^2 + y^2} \tag{B.4}$$

$$\theta = \tan^{-1}(y/x) \tag{B.5}$$

   (b) Transform the radius r to the distances l_f and l_b along the front and back spline surfaces using Equation (B.3). For example, point M in Figure B.1 would be mapped to the distance 0.8205 which is half the chord length of the spline curve.

   (c) Determine which segment of both the front and back spline curves contains this point and the proportional distance within this segment. The subroutine HUNT from Reference [2] is used to efficiently locate the segment. For example, point M' (the mapped point M) would be located in segment 4. The proportional distance within segment 4 is calculated as

$$\frac{0.8205 - 0.805}{1.223 - 0.805} = 0.0371$$

(d) Calculate the radii r_f and r_b which are the radii of the front and back faces corresponding to the distances l_f and l_b. For point M' this is:

$$r = 0.0371 \times (0.8 - 0.6) + 0.6 = .6074$$

where 0.6 and 0.8 are the radii at the beginning and end of segment 4.

(e) Calculate the Z-coordinates of the front and back surfaces at the radii r_f and r_b using the `SPLINT` (Spline Interpolation) routine. The Z-coordinate of point M is approximately 0.36.

(f) Calculate the X- and Y-coordinates corresponding to the radii r_f and r_b. If the original two-dimensional node coordinates are x and y, then

$$x_b = r_b \times x/\sqrt{x^2 + y^2} \quad (B.6)$$

$$y_b = r_b \times y/\sqrt{x^2 + y^2} \quad (B.7)$$

$$x_f = r_f \times x/\sqrt{x^2 + y^2} \quad (B.8)$$

$$y_f = r_f \times y/\sqrt{x^2 + y^2} \quad (B.9)$$

Note that these equations ensure that the generated three-dimensional node has the same polar angle θ in the X–Y plane as the original two-dimensional node.

(g) We now have the locations of the generated nodes on the front surface (x_f, y_f, z_f) and the back surface (x_b, y_b, z_b). Now, generate *ntran+1* nodes along this line with a gradient of *grad*.

Figures B.2 and B.3 show the result of this process. In Figure B.2, the two-dimensional mesh is a simple quarter-circle mesh with unit radius which is transformed into a eighth-spherical three-dimensional shell. Figure B.3 shows how a two-dimensional mesh is mapped when the mesh has a non-zero minimum dimension.

```
Spline 4 1
Angular
Front
   Left  0.0
   Right 0.0
   0.0   5.0
  45.0   5.0
  90.0   5.0
Back
   Left  0.0
   Right 0.0
   0.0   4.0
  45.0   4.0
  90.0   4.0
Exit
Exit
```

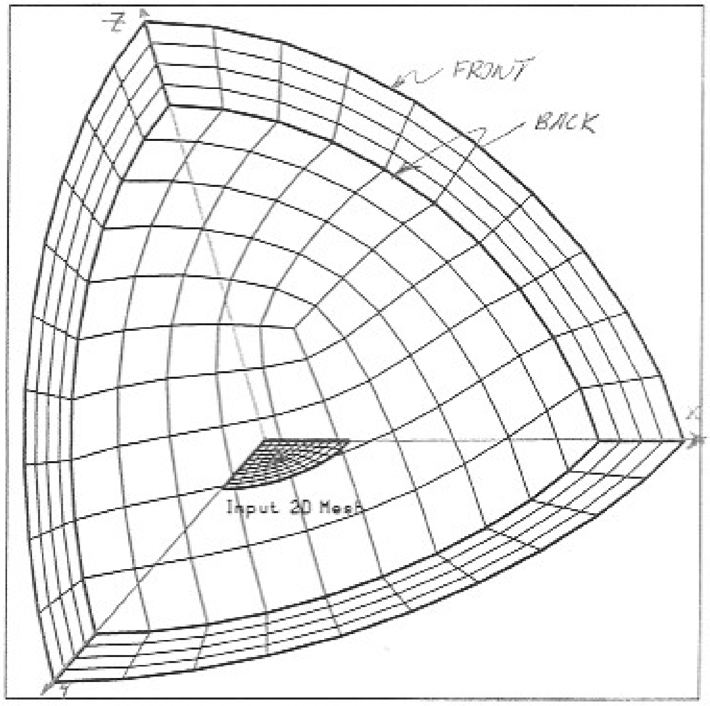

Figure B.2. Result of SPLINE transformation: quarter-circle two-dimensional mesh transformed into eighth-spherical three-dimensional shell.

```
SideSet Front 1
SideSet Back  2
Spline 4 1
Angular
Front
   Left  0.0
   Right 0.0
  45.0   5.0
  90.0   5.0
Back
   Left  0.0
   Right 0.0
  45.0   4.0
  90.0   4.0
Exit
Exit
```

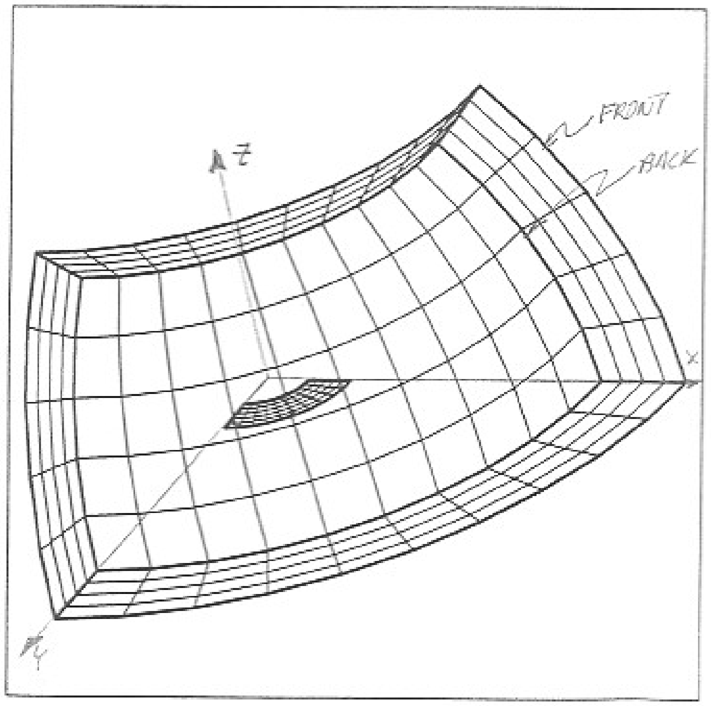

Figure B.3. Result of SPLINE transformation: two-dimensional mesh with hole transformed into three-dimensional shell mesh.