FLAC and Numerical Modeling in Geomechanics, Brummer et al. (eds)
© 2003 Swets & Zeitlinger, Lisse, ISBN 90 5809 581 9

An automated procedure for 3-dimensional mesh generation

A.K. Chugh
Bureau of Reclamation, Denver, CO, USA

T.D. Stark
University of Illinois, Urbana, IL, USA

ABSTRACT: An automated procedure is presented to generate a 3-dimensional mesh for numerical analysis of engineering problems. The procedure is simple, effective and efficient, and can be applied to represent complex geometries and material distributions. A listing of the program that was used for the sample problem of a landfill slide is included.

1 INTRODUCTION

One of the essential tasks in a 3-dimensional (3-D) numerical analysis is to represent the geometry and distribution of materials in the numerical model. $FLAC^{3D}$ provides means to facilitate mesh generation and the built-in programming language *FISH* can be used to develop and implement additional program instructions during execution of a data file.

In geotechnical engineering, surface geometry, distribution of materials, and water table conditions usually vary from one location to the next and pose a difficult set of conditions to represent in a numerical model. In order to facilitate the analysis of landslides, a simple procedure was devised to represent complex surface geometry, subsurface material horizons, and water table conditions. The objectives of this paper are to present:

1 a simple method to describe field geometry and conditions for a 3-D numerical model of a slope problem;
2 a simple procedure for automatic generation of a 3-D mesh; and
3 an illustration of the use of the procedure for analysis of a large slide in a landfill.

A listing of the program for the landfill slide is included in the paper. This program listing is in the *FISH* language and uses some of the functions available in the *FISH* library.

2 CONCEPTUAL MODEL

The conceptual model for the generation of a 3-D mesh follows the conventional procedure of portraying spatial variations of materials in 3-D via a series of 2-dimensional (2-D) cross-sections. This technique is commonly used by engineers and geologists in constructing visual models of complex geologic sites where a number of 2-D cross-sections are used to represent the field conditions. In these representations, linear variations between material horizons in consecutive 2-D cross-sections are used to depict the 3-D spatial variability of a site. The accuracy of the representation is improved by using closely spaced 2-D cross-sections.

The 3-D mesh generation procedure presented herein follows the conventional practices used by engineers in constructing 2-D numerical meshes by hand for geotechnical problems to be solved using methods other than $FLAC^{3D}$. For example, in the creation of a 2-D numerical model of a slope to be analyzed using a limit-equilibrium based procedure, it is a common practice to define profile lines via a set of data points followed by specifications of their connectivities. Also, in the creation of a 2-D model of a continuum to be solved by a finite-element based procedure, it is a common practice to discretize the continuum into a network of zones; assign identification numbers to the grid points; define the coordinates of the grid points; and then specify the connectivity of grid points.

Thus, in the conceptual model for the generation of a 3-D mesh in *FLAC*^{3D}, use is made of defining a series of 2-D cross-sections at representative locations of a site; defining each of the 2-D sections as an assemblage of data points with line-segment connections; and organizing the data for an efficient and effective discretization of the volume.

3 WATER TABLE

The water table surface is specified using the water table data of individual 2-D cross-sections and through the use of 3-point planar polygons between consecutive 2-D cross-sections. This scheme allows incorporation of non-coplanar variations in the water table surface in the entire 3-D model.

4 DESCRIPTION OF THE PROCEDURE

In geotechnical engineering, the ground-surface geo-metry is obtained using contour maps that are prepared from land or aerial survey of the area. The subsurface material horizons are estimated from geologic data and information obtained from exploratory boring logs. The subsurface water conditions are estimated from field observations, piezometers installed at various depths, and/or from water levels in borings. Subsurface data are used to develop contour maps of the subsurface geology and water conditions.

From these contour maps, the region-of-interest, and the locations of significant cross-sections are identified; information for 2-D cross-sections are read and tabulated; and 2-D cross-sections are drawn for an understanding of the site details and preparation of input data for a 2-D analysis. In general, the cross-sectional data for a site varies from one location to the next. These variations may be caused by changes in the ground surface and (or) in subsurface material horizons, discontinuity of some materials, or a combination of these or some other variations.

In the proposed procedure, the following steps are followed: (For ease of presentation, 2-D cross-sections are assumed to lie in x-z plane and the x,y,z coordinate system follow the right hand rule.)

1 The following steps are used for creating an orderly assemblage of field data for 3-D discretization of the continuum of the region-of-interest:

   a On the site map, select values of x, y, and z coordinates that completely circumscribe the 3-D region-of-interest;

   b Mark locations of all significant 2-D cross-sections oriented in the same and preferably parallel direction;

   c For each 2-D cross-section, tabulate (x,y,z) coordinates of end-points of all line segments for each profile line and the water table (for parallel 2-D cross-sections, y-coordinate shall have same constant value between two consecutive cross-sections).

2 The following steps are used for creating similar sets of data at each of the 2-D cross-sections:

   a From the data in step 1(c) above, select control points that are of significance in defining the profile lines in all of the 2-D cross-sections. Tabulate the x-coordinates of these control points in increasing order. For reference purposes, this table is referred to as Table 100.

   b Use of the "Interpolate" function expands the 2-D cross-sectional data of step 1(c) by linear interpolation for all of the control points listed in Table 100 for all of the profile lines and stores the data in separate tables; assigns Table numbers in increasing order starting with the user specified starting number and incrementing it by 1; assigns an identification number to each point; and positions the points in the 3-D model space. These tables contain the (x,z) coordinates of expanded 2-D cross-sectional data. A sample listing of the "Interpolate" function and its dependency function "zz" in *FISH* language is given in Figure 1. The starting table number used in the sample problem data file is 200.

3 The following steps are used for creating zones in the 3-D model space:

```
def zz
zz=table(t_n,xx)
end

def interpolate
loop j (js,je); profile line #s -
; js is for the bottom, je is for top
dt_n=dt_n_s+j; dt_n is destination table number
loop i (is,ie); is is the first interpolation #,
; ie is the last interpolation #
xx=xtable (100,i); x-coordinate of the
;interpolation point
command
set t_n=j
end_command
table(dt_n,xx)=zz
id_pt=id_pt+1
x_pt=xtable(dt_n,i)
y_pt=y_pt
z_pt=ytable(dt_n,i)
command
generate point id id_pt x_pt y_pt z_pt
end_command
endloop
endloop
end
```

Figure 1. Listing of the "Interpolate" function and its dependency function "zz" in *FISH* language.

a Tabulate the y-coordinates of the 2-D cross-sections in increasing y-direction. For reference purposes, this table is referred to as Table 101. The number of entries in Table 101 should equal the number of 2-D cross-sections marked in step 1(b).

b Considering the spacing of x-coordinates of the control points in step 2(a), select the number of zones desired for each interval in the x-direction. Tabulate these values for all of the intervals in the increasing x-direction. For reference purposes, this table is referred to as Table 102. The number of entries in Table 102 should be one less than those in Table 100.

```
def fill_grid
i_n=table_size(102)
j_n=table_size(103)
k_n=table_size(104)
loop jy (1,j_n)
ny=xtable(103,jy)
p0_d=(jy-1)*(i_n+1)*(k_n+1)
loop kz (1,k_n)
nz=xtable(104,kz)
if kz=1 then
material='shale'
endif
if kz=2 then
material='ns'; native soil
endif
if kz=3 then
material='msw'; municipal solid waste
x_toe=xtable(105,jy)
endif
loop ix (1,i_n)
if kz=3 then
xx_toe=xtable(100,ix)
if xx_toe < x_toe then
material='mswt'
endif
endif
nx=xtable(102,ix)
p0_d=p0_d+1
p3_d=(p0_d+i_n+1)
p6_d=(p3_d+1)
p1_d=(p0_d+1)
p2_d=((i_n+1)*(k_n+1)+p0_d)
p5_d=(p2_d+(i_n+1))
p7_d=(p5_d+1)
p4_d=(p2_d+1)
command
generate zone brick size nx,ny,nz ratio 1,1,1 &
p0=point (p0_d) p3=point (p3_d) &
p6=point (p6_d) p1=point (p1_d) &
p2=point (p2_d) p5=point (p5_d) &
p7=point (p7_d) p4= point(p4_d) group material
end_command
if kz=3 then
material='msw'
endif
end_loop
p0_d=p0_d+1
end_loop
end_loop
end
```

Figure 2. Listing of "FILL_GRID" function in *FISH* language.

c Considering the spacing between the 2-D cross-sections in the y-direction, select the number of zones desired for each interval in the y-direction. Tabulate these values for all of the intervals in the increasing y-direction. For reference purposes, this table is referred to as Table 103. The number of entries in Table 103 should be one less than the number of 2-D cross-sections.

d Considering the spacing of the profile lines in the z-direction, select the number of zones desired for each material horizon in the z-direction. Tabulate these values for all of the intervals in the increasing z-direction. For reference purposes, this table is referred to as Table 104. The number of entries in Table 104 should be one less than the number of profile lines.

e Use of the "Fill_grid" function generates a brick mesh and assigns a group name to each 3-D volume zone. A sample listing of the "Fill_grid" function in *FISH* language is given in Figure 2.

5 COMMENTS

1 Use of a Brick mesh with an 8-point description is versatile and allows for creation of degenerated brick forms through the use of multiple points with different identification numbers occupying the same (x,y,z) coordinate location in the 3-D model space.

2 During the development of the grid, it is possible to assign group names to different segments of the model. This information can be useful in modifying the generated grid.

3 Expanding the (x,y,z) location data for all 2-D cross-sections to a common control number of locations via interpolations facilitates the programming of the automatic grid-generation procedure.

4 In engineering practice, it is generally desirable to analyze a few 2-D cross-sections at select locations prior to conducting a 3-D analysis. Because development of data for 2-D cross-sections is one of the steps for use of the proposed procedure, it is relatively easy to conduct a 2-D analysis using the 2-D cross-sectional data and the program *FLAC*.

5 The program instructions listed in Figures 1 and 2 can be modified to accommodate geometry and other problem details that are different or more complex than those encountered in the sample problem described in Section 6.

6 SAMPLE PROBLEM

The problem used to illustrate the proposed 3-D mesh generation procedure is the 1996 slide in a waste containment facility near Cincinnati, Ohio (Stark & Eid 1998, Eid et al. 2000). Figure 3 is an aerial view of the slide. Figure 4 is the plan view of the landfill and shows the location of the sixteen cross-sections used to construct a *FLAC*^{3D} model of the site (the project data shown are in Imperial units). There are three material horizons bounded by four profile lines, and a liquid level present at this site. Figure 5 shows the 2-D cross-sectional views of the site at the 16-locations prior to failure (the available project data were converted to SI units and this conversion lead to numerical values with fractional parts). Figure 6 shows a partial listing of the data file for the sample problem with the following details:

- Table 100 lists the x-coordinates of the 22 control points considered significant from the sixteen 2-D cross-sectional data.
- Table 101 lists the y-coordinates of the sixteen 2-D cross-section locations.
- Table 102 lists the number of zones desired in each of the 21 segments in the x-direction.
- Table 103 lists the number of zones desired in each of the 15 segments in the y-direction.
- Table 104 lists the number of zones desired in each of the 3 material horizons at the site.
- Table 105 lists the x-coordinates of the toe locations of the top profile line in the 2-D cross-sections in the increasing y-direction.

For each cross-section, x- and z-coordinates for data points defining the profile lines are recorded in individual tables numbered as Table 1 for profile line 1 data, Table 2 for profile line 2 data, Table 3 for profile line 3 data, and Table 4 for profile line 4 data in the data file shown in Figure 6. Profile lines are numbered from 1 to 4 in the increasing z-direction and each profile line uses a different number of data points to define the line. For cross-sections where the top profile line terminates in a vertical cut at the toe, the top profile line was extended to x = 0.

For each cross-section and for each of the four profile lines, the x-coordinate locations identified in Table 100 are used to create data by interpolation at each of the 22 control points. For the sample problem, this amounts to 88 pairs of (x,z) coordinates per cross-section, and the y-coordinate of the data points is read from Table 101. Thus, the x-,y-, and z-coordinates for all of the points defined and (or) interpolated are known. Each point is assigned a numeric identity number (id #) starting with one and incrementing by one. The data points are located in the 3-D model space using their id # and x-,y-, z-coordinates. This task is accomplished using the "Interpolate" function and its listing in *FISH* language is given in Figure 1. At the end of this task, all of the defined and (or) interpolated points with an assigned id # have been located in the 3-D model space.

The connectivity of data points to define volume discretization is accomplished in the function named

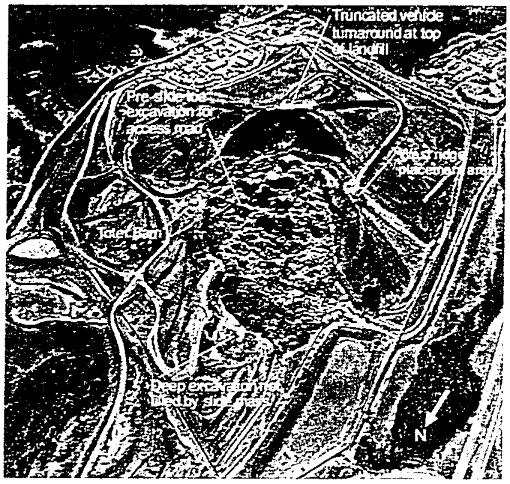


Figure 3. Sample problem – aerial view of Cincinnati landfill failure (from Eid et al. 2000). (Reproduced by permission of the publisher, ASCE).

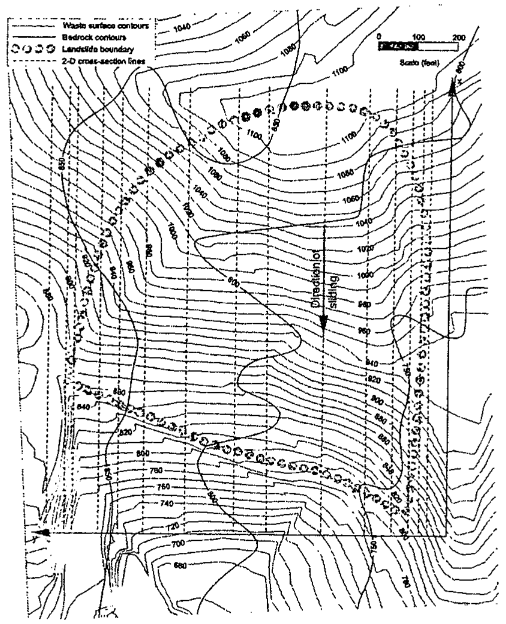
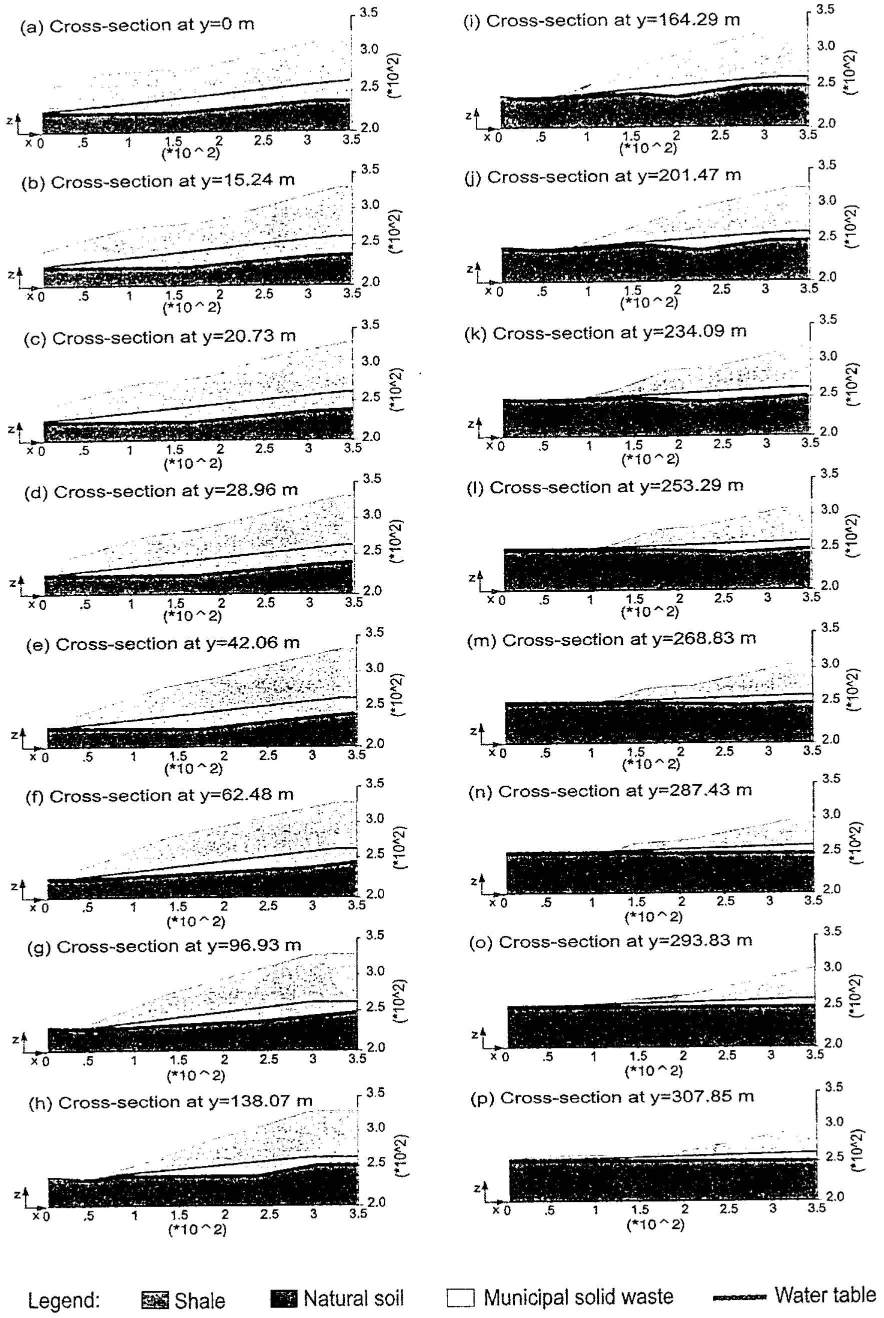


Figure 4. Plan view of the sample problem showing locations of selected 2-D sections.


(a) Cross-section at y=0 m

(b) Cross-section at y=15.24 m

(c) Cross-section at y=20.73 m

(d) Cross-section at y=28.96 m

(e) Cross-section at y=42.06 m

(f) Cross-section at y=62.48 m

(g) Cross-section at y=96.93 m

(h) Cross-section at y=138.07 m

(i) Cross-section at y=164.29 m

(j) Cross-section at y=201.47 m

(k) Cross-section at y=234.09 m

(l) Cross-section at y=253.29 m

(m) Cross-section at y=268.83 m

(n) Cross-section at y=287.43 m

(o) Cross-section at y=293.83 m

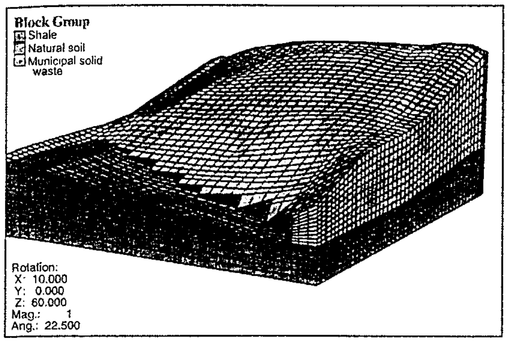
(p) Cross-section at y=307.85 m

Figure 5. 2-D cross-sectional views of the sample problem.

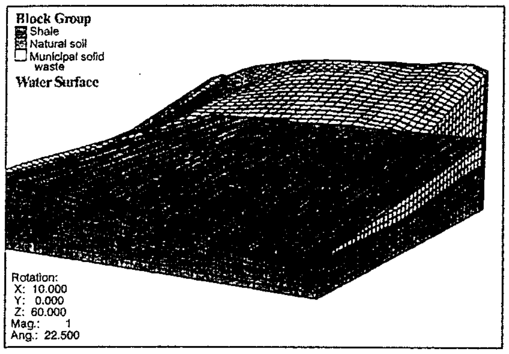
“Fill_grid”. For each interval in the location of cross-sections in the y-direction (Table 103), and for each material horizon between the profile lines in the z-direction (Table 104), and for each interval in the x-direction (Table 102), the values of number of zones desired in the x, y, and z-direction and the id #s of points in the 3-D model space are used in the “**GENERATE zone brick p0, p1, … p8**” command of

```
; Rumpke landfill site; Data are in metric units
set g=0,0,-9.81

; table 100 is for the x-coordinates of
; the desired 3-D grid
table 100 0,1 13.11,2 15.54,3 22.86,4 34.75,5
table 100 42.67,6 49.07,7 57.61,8 63.70,9
table 100 64.92,10 72.54,11 78.94,12 92.66,13
table 100 100.89,14 107.90,15 115.21,16
table 100 158.50,17 199.64,18 284.38,19
table 100 318.52,20 337.72,21 348.08,22

; table 101 is for y-coordinates of the
; 2-D cross-section locations
table 101 0,1 15.24,2 20.73,3 28.96,4 42.06,5
table 101 62.48,6 96.93,7 138.07,8 164.29,9
table 101 201.47,10 234.09,11 253.29,12
table 101 268.83,13 287.43,14 293.83,15
table 101 307.85,16

; table 102 is for the number of zones
; desired in the x-direction
table 102 2,1 1,2 1,3 2,4 1,5 1,6 1,7 1,8 1,9
table 102 1,10 1,11 2,12 1,13 1,14 1,15 5,16
table 102 5,17 10,18 4,19 2,20 2,21

; table 103 is for the number of zones
; desired in the y-direction
table 103 2,1 1,2 1,3 2,4 2,5 3,6 4,7 3,8 4,9
table 103 3,10 2,11 2,12 2,13 1,14 2,15

; table 104 is for the number of zones
; desired in the z-direction
table 104 5,1 3,2 10,3

; table 105 is for the x-coordinates of the
; receding toe
table 105 0,1 0,2 15.54,3 22.86,4 34.75,5
table 105 49.07,6 57.61,7 64.92,8 78.94,9
table 105 92.66,10 100.89,11 107.90,12
table 105 115.21,13 63.70,14 0,15

set is=1 ie=22
set js=1 je=4
set id_pt=0
set dt_n_s=200

; Station at y=0
set y_pt=0
table 1 -100,200 500,200
table 2 0,223.60 154.23,223.60 307.24,238.84
table 2 348.08,239.14
table 3 0,228.60 154.23,228.60 307.24,243.84
table 3 348.08,244.14
table 4 0,260.00 66.45,280.42 98.15,283.46
table 4 156.67,286.51 187.15,289.56
table 4 348.08,332.54
interpolate

; station at y=15.24 m
set y_pt=15.24
table 2 erase
table 3 erase
table 4 erase
set dt_n_s=dt_n
table 2 0,223.60 163.07,223.60 306.02,238.84
table 2 348.08,240.67
table 3 0,228.60 163.07,228.60 306.02,243.84
table 3 348.08,245.67
table 4 0,251.46 91.14,280.42 107.90,283.46
table 4 144.48,286.51 169.77,289.56
table 4 194.46,292.61 332.54,338.33
table 4 348.08,338.33
interpolate
.
.
.
; station at y=307.85 m
set y_pt=307.85
table 2 erase
table 3 erase
table 4 erase
set dt_n_s=dt_n
table 2 0,254.08 348.08,254.08
table 3 0,259.08 348.08,259.08
table 4 0,261.08 29.87,265.18 185.93,268.22
table 4 348.08,307.24
interpolate

fill_grid

delete range group mswt

; water surface
water den=1 table &
face 0,0,228.60 0,15.24,228.60 &
332.54,15.24,268.22 &
face 0,0,228.60 332.54,15.24,268.22 &
348.08,15.24,268.22 &
face 0,0,228.60 348.08,15.24,268.22 &
348.08,0,268.22 & ;interval # 1
face 0,15.24,228.60 0,20.73,228.60 &
340.77,20.73,268.22 &
face 0,15.24,228.60 340.77,20.73,268.22 &
348.08,20.73,268.22 &
face 0,15.24,228.60 348.08,20.73,268.22 &
332.54,15.24,268.22 &
face 332.54,15.24,268.22 348.08,20.73,268.22 &
348.08,15.24,268.22 &;interval # 2
.
.
.
face 0,293.83,259.08 0,307.85,259.08 &
63.70,307.85,259.08 &
face 0,293.83,259.08 63.70,307.85,259.08 &
348.08,307.85,268.22 &
face 0,293.83,259.08 348.08,307.85,268.22 &
63.70,293.83,259.08 &
face 63.70,293.83,259.08 348.08,307.85,268.22 &
348.08,293.83,268.22;interval # 15

save cin_3D_grid.sav
```

Figure 6. Partial listing of the data file for the sample problem for *FLAC*^{3D}.



(a) - without the water surface



(b) with the water surface

Figure 7. 3-D mesh for the sample problem.

FLAC^{3D} for a regular 8-noded brick mesh. The material between the profile lines is assigned a group name for ease of modifying the grid and for convenience in assigning material properties and/or addressing them for some other reason. This task is also accomplished in the function named "Fill_grid" and its listing in *FISH* language is given in Figure 2. Table 105 data are used to assign a group name "mswt" to the zones past the vertical cut which are later deleted using the **DELETE** command with the **range** defined by the **group** name "mswt". At the end of this task, a 3-D grid of specification exists in the region-of-interest. For the sample problem, the generated 3-D grid is shown in Figure 7. The representation of continuity of the vertical cut at the toe of the slope (as seen in 2-D cross-sections, Figure 5) in the 3-D model can be improved by increasing the number of 2-D cross-sections.

7 ADVANTAGES OF THE PROPOSED PROCEDURE

1 The proposed procedure for describing 3-D field conditions utilizes 2-D cross-sections, which are essentially the same as commonly used by geologists and engineers to describe the field conditions. Linear variation in geometry, material horizons, and groundwater descriptions between known data points is generally accepted.
2 Changes in field data can be incorporated in the numerical model by updating the affected tables.
3 New cross-sections can be introduced or old cross-sections deleted and a new discretization of the continuum made quickly.
4 Describing the spatial location of data in a 3-D space followed by descriptions of their connectivity is a simple yet powerful way of constructing a 3-D numerical model for analysis purposes.
5 The proposed procedure produces regions with acceptable geometries, i.e. no conflicts in connectivity.
6 Changes in discretization due to changes in field data or due to numerical considerations can be included in the proposed procedure efficiently and a new discretization accomplished.
7 Number of discretized volume units in different parts of the numerical model is estimated at the start of the problem solving effort. If it becomes necessary to change or refine the discretization, very little effort is needed to change the tabular data and the procedure is then rerun to obtain an updated 3-D mesh.
8 A complete brick element is used to generate other degenerated volume element shapes.
9 Because the proposed procedure is based on simple and commonly used ideas, it should be adaptable when using computer programs or procedures other than *FLAC^{3D}* to perform numerical analysis work. The program instructions can be rewritten in other programming languages.

8 SUMMARY

To facilitate 3-D analyses using *FLAC^{3D}* or other software, an automated procedure is presented to create a 3-D mesh. The procedure utilizes commonly used techniques for drawing 2-D cross-sections and interpolation between 2-D cross-sections to portray spatial variations of geometry and distribution of materials in 3-D.

REFERENCES

Eid, H.T., Stark, T.D., Evans, W.D. & Sherry, P.E. 2000. Municipal solid waste slope failure. II Stability analyses. *Journal of Geotechnical and Geoenvironmental Engineering* 126(5): 408–419.

Stark, T.D. & Eid, H.T. 1998. Performance of three-dimensional slope stability methods in practice. *Journal of Geotechnical and Geoenvironmental Engineering* 124(11): 1049–1060.