SCHOOL OF
CIVIL ENGINEERING
INDIANA
DEPARTMENT OF HIGHWAYS

Joint Highway Research Project

Final Report

FHWA/IN/JHRP-91/7

USER GUIDE FOR PC-STABL3D

Gennady Moshkovich

PURDUE UNIVERSITY

Joint Highway Research Project

Final Report

FHWA/IN/JHRP-91/7

USER GUIDE FOR PC-STABL3D

Gennady Moshkovich

Final Report

USER GUIDE FOR PC-STABL3D

To: H. L. Michael, Director
Joint Highway Research Project

March 6, 1991
Revised April 23, 1991
Project No: C-36-36N
File No: 6-14-14

From: C. W. "Bill" Lovell
Professor of Civil Engineering
Purdue University,
West Lafayette, Indiana

Subj: User Guide for PC-STABL3D

Attached is the final report on the subject study. The report has been prepared by Mr. Gennady Moshkovich, Graduate Research Assistant, under my direction.

The results of this report will allow the INDOT to analyze three-dimensional slope stability problems with a three-dimensional model, instead of a two-dimensional one.

The program PC-STABL3D has been revised to reduce "bugs" in the previous version of the program, and to make the program "user friendly". The report is, in fact, a Users' Manual for INDOT personal.

The report is submitted for review, comment and acceptance in fulfillment of the referenced study.

Respectfully Submitted

C.W. Lovell

cc.

FINAL REPORT

USER GUIDE FOR PC-STABL3D

by

Gennady Moshkovich
Graduate Research Assistant

Joint Highway Research Project

Project No: C-36-36N
File No: 6-14-14

Prepared as Part of an Investigation
Conducted by the
Joint Highway Research Project
Engineering Experiment Station
Purdue University

In Cooperation with the
Indiana Department of Transportation

and the

U.S. Department of Transportation
Federal Highway Administration

The content of this report reflects the views of the author who is responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Indiana Department of Transportation. This report does not constitute a standard, specifications or regulations.

Department of Civil Engineering
Purdue University
West Lafayette, Indiana

March, 1991
Revised April 23, 1991

1. Report No. FHWA/INJHRP-91/7	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle User Guide for PC-STABL3D		5. Report Date: March 6, 1991. Revised April 23, 1991.	
		6. Performing Organization Code	
7. Author(s) Gennady Moshkovich		8. Performing Organization Report No. FHWA/INJHRP-91/7	
9. Performing Organization Name and Address Joint Highway Research Project Civil Engineering Building Purdue University West Lafayette, IN 47907		10. Work Unit No.	
		11. Contract or Grant No.	
12. Sponsoring Agency Name and Address Indiana Department of Transportation State Office Building 100 North Senate Avenue Indianapolis, IN 46204		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract PC-STABL3D is the program for IBM PC compatible computers, developed to handle general stability analysis of three-dimensional slopes. Random generation routines are invoked to create trial failure surfaces, which are analyzed using the limiting equilibrium approach. The current version of PC-STABL3D is enhanced by a graphical pre/postprocessor, which facilitates creation and edition of data files, examination of problem geometry, and analyzing of the obtained failure surfaces. After the failure surfaces are generated and their factors of safety are computed, the program performs statistical analysis of received factors of safety, and calculates minimum statistical factor of safety. The information about minimum factor of safety increases probability that the critical failure surface found by the program is indeed the one with minimum factor of safety. The manual also contains a step-by-step tutorial and example problems which are solved using PC-STABL3D. The contents of this report summarize a series of previous research efforts relating to three-dimensional slope stability, conducted under the guidance of Prof. C. W. Lovell and Prof. J-L. Chameau.			
17. Key Words slope stability, three-dimensional, computer modeling, statistical factor of safety		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 250	22. Price

Digitized by the Internet Archive
in 2011 with funding from
LYRASIS members and Sloan Foundation; Indiana Department of Transportation

ACKNOWLEDGMENTS

I wish to express my sincere gratitude and thanks to Professor C. W. Lovell for his guidance and attention during the time of this research. Thanks are also extended to Professor D. Frost who spent his valuable time to share his ideas and to offer his moral support.

My thanks to William Sisiliano and Nadim Raza from the Indiana Department of Transportation for their suggestions on improvement of the program.

Financial support for this project was provided by the INDOT and the Federal Highway Administration through the Joint Highway Research Project, School of Civil Engineering, Purdue University, West Lafayette, Indiana.

I am deeply thankful to all my friends who helped me during the entire time at Purdue University.

ABSTRACT: PC-STABL3D is the program for IBM PC compatible computers, developed to handle general stability analysis of three-dimensional slopes. Random generation routines are invoked to create trial failure surfaces, which are analyzed using the limiting equilibrium approach. The factors of safety predicted by the program are statistically analyzed using the Weibull's distribution and the minimum statistical factor of safety is calculated as a result of this analysis. The information about minimum statistical factor of safety may help user to evaluate reliability of the obtained critical failure surface. The current version of PC-STABL3D is enhanced by a graphical pre/postprocessor, which facilitates creation and edition of data files, examination of problem geometry, and analyzing of the obtained failure surfaces. The manual also contains a step-by-step tutorial and example problems which are solved using PC-STABL3D. The contents of this report summarize a series of previous research efforts relating to three-dimensional slope stability, conducted under the guidance of Prof. C. W. Lovell and Prof. J-L. Chameau.

TABLE OF CONTENTS

1.0	INTRODUCTION	1
2.0	PROBLEM GEOMETRY	2
2.1	Soil Boundaries	5
2.2	Definition of Searching Boundaries . .	5
2.2.1	Definition of the Main Sliding Axis	7
2.2.2	Ground Surface Boundary	8
2.2.2.1	Limits in Y Direction . .	8
2.2.2.2	Limits in X Direction . .	8
2.2.2.3	Boundaries Within the Soil Mass	13
3.0	SOIL PARAMETERS AND GROUND WATER	14
3.1	Soil Parameters	14
3.2	Ground Water	14
4.0	DATA PREPARATION	15
5.0	GRAPHICAL PRE/POSTPROCESSOR	20
5.1	Overview	20
5.2	Creation and Edition of Data Files . .	22
5.3	Examining of the Surface Geometry . . .	34
6.0	MISCELLANEOUS INFORMATION	40
6.1	Modification and Revisions of PC-STABL3D	40
6.2	Units	41
6.3	Problem Size Limitations	42
7.0	RESULTS	42
8.0	CONCLUSIONS	42a
	REFERENCES	43
	APPENDICES	44
	APPENDIX A STATISTICAL ANALYSIS	45
	APPENDIX B EXAMPLES	49
	APPENDIX C PROGRAM SOURCE CODE	128
	APPENDIX D NEGATIVE NUMBERS FOR PRINTS .	250

LIST OF FIGURES

Figure 1	System of Coordinates (After Thomaz, 1986)	3
Figure 2	Grid of Cross Sections (After Thomaz, 1986)	4
Figure 3	Main Sliding Axis of the Cross Section	6
Figure 4	Plan View of the Slope	8
Figure 5	Definition of Limits in Y Direction	10
Figure 6	Definition of Limits in X Direction	11
Figure 7	Definition of Limits in Z Direction	12
Figure 8	Main Menu	21
Figure 9	Prompt for Name of Data File	23
Figure 10	Prompt for the Number of Grid Points	24
Figure 11	Prompt for the Horizontal Coordinates of Nodes	25
Figure 12	Prompt for the Vertical Coordinates of Nodes	26
Figure 13	Prompt for the Failure Surface Generation Data	28
Figure 14	Prompt for the Initiation Values of Surface Generation	29
Figure 15	Prompt for the Soil Properties	30
Figure 16	Prompt for the Pore Pressure Model	31
Figure 17	Prompt for the Coordinates of Piezometric Surfaces	33
Figure 18	Pop-up File Listing	35
Figure 19	Prompt for the Cross Section Number	36
Figure 20	Example of a Cross Section	37
Figure 21	Example of a Contour Plot	39
Figure B1	Illustration for Example 1	49
Figure B2	Illustration for Example 2	69
Figure B3	Illustration for Example 3	90
Figure B4	Illustration for Example 4	109

1.0 INTRODUCTION

PC-STABL3D is a computer program written in FORTRAN 77 computer language for the general solution of three dimensional slope stability problems. It is compiled using a Microsoft FORTRAN 4.1 (R)¹ compiling package. The calculation of the factor of safety against the instability of a slope is performed by the general three-dimensional method of columns developed at Purdue University (Chen, 1981; Thomaz, 1986).

PC-STABL3D features unique random techniques for generation of potential failure surfaces for subsequent determination of the most critical surface and its corresponding factor of safety. The surfaces are generated within limits determined by the user.

PC-STABL3D is supplied with a graphical pre/postprocessor to facilitate and visualize the input/output process. The pre/postprocessor includes a simple editor developed for the creation and editing of the PC-STABL3D data files. Graphical capabilities of the pre/postprocessor allow the user to check the correctness of the problem geometry and to view resulting failure surfaces. The graphical pre/postprocessor is written in the QuickBASIC 4.5 (R)² computer language.

The program performs statistical analysis of received factors of safety. The value of the minimum statistical factor of safety is calculated as a result of this analysis. The analysis is performed using the extreme value distribution for minimums. This distribution is also called Weibull's distribution. The theoretical background for this analysis is presented in Appendix A.

---

¹ Microsoft FORTRAN 4.1 is a registered trademark of Microsoft Corporation.

² QuickBASIC 4.5 is a registered trademark of Microsoft Corporation.

Appendix B contains examples of the input and output files. These examples show capabilities of PC-STABL3D to predict factor of safety for concave in plan slopes. The time to run these examples with 500 iterations using PC 486 was an average of 4-5 minutes. The source code of the program can be found in Appendix C. The first part is the source code of the main program, and the second is the source code of the graphical pre/postprocessor.

The minimum memory required for the program is 640kB. Though the program is available for computers with and without a math coprocessor, it is advisable to run the program with a math coprocessor to increase the speed of calculations.

This manual is not intended to totally explain how PC-STABL3D operates or what assumptions are made to arrive at a solution. For a more detailed explanation of the logical operation of PC-STABL3D and mathematical models employed refer to Thomaz (1986) and Chen (1981).

2.0 PROBLEM GEOMETRY

To begin, the user must plot the problem geometry to scale on a rectangular coordinate grid. Due to the three-dimensional nature of the problem, the following procedure is suggested in order to prepare data points for the program. The system for representing the geometry of the problem will consist of a three-dimensional cartesian system. The user must define three orthogonal axes, X, Y, and Z. The horizontal plane should be defined by axes X and Y; then the Z will represent the elevations of the layers and ground profiles. During the determination of this coordinate system the user should remember that the direction of the slides being investigated is along the Y axis. Figure 1 shows the representation of the sliding axes described above.

The next step consists of defining the points (coordinates) that will represent the geometry. The way these coordinates are to be defined is shown on Figure 2. The user will need to define a grid consisting of cross sections of the slope parallel to the X and to the Y axes. The spacings between these

Figure 1 System of Coordinates (After Thomaz, 1986)

Figure 2 Grid of Cross Sections (After Thomaz, 1986)

cross sections do not need to be the same, nevertheless, it is highly recommended that the variations in the spacing are minimal.

The horizontal projection of the resulting mesh can be seen as a series of axes in the X direction and another series in the Y direction. The intersections of this mesh are the points that will define the geometry of the problem on the horizontal surface.

These points can be perceived as locations of boreholes. For each of the points, the user must define the elevations of the ground surface, as well as the elevations of the different soil layers and piezometric surfaces.

2.1 Soil Boundaries

After the grid has been prepared, the user needs to define soil boundaries by specifying the elevations of the different soil layers at each of the nodes of the grid. The resulting profile will be represented by line segments connecting the nodes of the corresponding soil layer for each cross section in the X and Y directions (Figure 3). The position of the cross sections should be selected so that no important feature of the subsoil is incorrectly represented. To simplify positioning of cross sections the user can use the pre/postprocessor, which is described later in this manual.

2.2 Definition of Searching Boundaries

The appropriate choice of the searching boundaries can considerably optimize the surface searching process and hence reduce the amount of time involved in the search by avoiding the generation of "useless" surfaces. Useless surfaces are those with little or no probability of being

Figure 3 Main Sliding Axis of the Cross Section

critical (surfaces passing through sound rock, for instance). Some engineering judgment is thus required when defining the region to be searched.

It should be noted that even if the boundaries are very wide, the resulting critical region will still be the same. Nevertheless, the amount of time necessary to arrive at this final region will be much higher, since a higher percentage of the generated surfaces will probably lie outside of the critical region of the slope.

Definition of the boundaries consists of three steps:

- Definition of the main sliding axis;
- Definition of the region on the ground surface within which the surfaces are predicted to initiate and to terminate;
- Definition of the region within the soil mass which is to be processed by the program.

2.2.1 Definition of the Main Sliding Axis

The program requires the creation of a main sliding axis, which is to be defined parallel to the Y axis as it is shown on Figure 4. The main sliding axis is actually one of the cross sections already defined above. The location for this cross section should be chosen in a way so that it will lie along or as close as possible to the axis of symmetry of the sliding mass. For instance, if the user previously defined 11 cross sections in the Y direction, the best choice for the main sliding axis would be the cross section number 6. It is very important to select an appropriate cross section for the main sliding axis, because it will define initial points for failure surface generation.

Figure 4 Plan View of the Slope

2.2.2 Ground Surface Boundary

The ground surface boundary is the region where the intersection line between failure surfaces and the ground surface is located. The best way to understand how this boundary is established is to consider a plan view of the slope (Figure 5). The previously-defined cross sections now appear as a mesh, and intersections between failure surfaces and the ground surface will be represented by the polygon. Now the user will have to define boundaries for this polygon in the X and Y directions.

2.2.2.1 Limits in Y Direction

The user must define limits in the Y direction on the main axis of sliding (Figure 6). To establish the region within which the initial points of the generated surface are expected to lie, define a minimum value $Y_{i\min}$ and a $\Delta Y_{i\min}$. The trial surfaces will be initialized only from points located in the region between coordinates $Y_{i\min}$ and $Y_{i\min} + \Delta Y_{i\min}$. Similarly the user must define the region on the main sliding axis for termination of the trial surfaces, which will be contained between coordinates $Y_{f\min}$ and $Y_{f\min} + \Delta Y_{f\min}$. After the limits in the Y direction are defined, the user will need to define limits in the X direction.

2.2.2.2 Limits in X Direction

The procedure to define the limits in the X direction is basically the same as that for the Y direction, but now these regions must be defined for each of the cross sections in the X direction (Figure 7). The user will define the leftmost coordinate of the surface initiation $X_{i\min}$ with the corresponding surface initiation interval $\Delta X_{i\min}$, and the leftmost coordinate of the surface termination $X_{f\min}$ with the corresponding surface termination interval $\Delta X_{f\min}$.

Figure 5 Definition of Limits in Y Direction

Figure 6 Definition of Limits in X Direction

Figure 7 Definition of Limits in Z Direction

If these values were defined correctly, the failure surface will be initiated in the region from $X_{i \min}$ to $X_{i \min} + \Delta X_{i \min}$, and terminated in the region from $X_{f \min}$ to $X_{f \min} + \Delta X_{f \min}$.

2.2.2.3 Boundaries Within the Soil Mass

After the user has determined the surface boundaries of the sliding mass, the region inside the soil mass where the failure surfaces are to be generated must be defined.

The procedure used to define the region for surfaces generation within the soil mass (in Z direction) is very similar to that used to define the surface boundaries. For each node of the mesh the user will define the minimum coordinate $Z_{\min}$ and the distance $\Delta Z_{\min}$. The minimum coordinate $Z_{\min}$ is the maximum depth of failure surface generation, the coordinate $Z_{\min} + \Delta Z_{\min}$ is the minimum depth of the failure surface generation.

Boundaries within the soil mass can be used to limit surface generation to a specified region, for instance, when the soil profile contains weak layers, or sound rock. It is also helpful to make several computer runs of the program progressively optimizing the region to be searched, in order to achieve the most reliable solution.

The routine for surface generation contains automatic checks and numerical bias to control the shapes of the surfaces; however the imposed search limit boundary restrictions always have precedence over any other methodology used in the program.

3.0 SOIL PARAMETERS AND GROUND WATER

3.1 Soil Parameters

Each soil type is described by the following set of isotropic parameters: the moist or saturated unit weight γ , the Mohr-Coulomb strength intercept c (cohesion), and the Mohr-Coulomb strength angle ϕ . The program performs calculations using the total weight and total boundary force approach. However with the measurement or prediction of pore pressure, either a total stress or an effective stress analysis may be undertaken. Moist unit weight is assumed above the water level; saturated unit weight below it.

3.2 Ground Water

The ground water conditions can be represented using one of the two descriptors: coordinates of the piezometric surfaces and values of the pore pressure ratio. The former is most commonly applied.

NOTE: When there is no ground water present, water conditions are described as having a pore pressure ratio equal to zero, and γ is the moist unit weight.

In the two dimensional program (PC STABL 5M), the water pressure produced by a piezometric surface may be defined by three options. In the current program only the simple equivalent hydrostatic value is used. The user will need to define piezometric surfaces involved in the analysis in the same way as for the soil layers. For each point of the mesh, the elevation corresponding to each of the piezometric surfaces must be described.

The piezometric surfaces may lie anywhere within the region being defined. They can be above the ground surface or coincide with it at different points. If more than one piezometric surface is defined and one of them represents a body of water, such as lake or river, it should be assigned to the first layer of the soil. The program will check if the first piezometric surface lies above the ground, and wherever it does, it will add the weight of water to the calculations of the total weight of the column. If a piezometric surface which is located above the ground surface, is not the one assigned to the top layer, the water will be considered for pore pressure calculation only.

The pore pressure ratio r_u is the ratio of the pore pressure to the vertical total stress at the given elevation.

4.0 DATA PREPARATION

The user can create input files for the PC-STABL3D program using either the supplied pre/postprocessor or a conventional editor. In both cases the format of the data file will be the same. Therefore the input file format, described in this section, is applicable to files created with both the conventional editor and the pre/processor. In order to simplify generation of data file with an editor, all data used in the program are read in the free-form format; hence users do not need to be concerned about missing periods or extra blanks between numbers.

The format of the data file is described below. A new line should be started whenever a DATA CARD statement is encountered.

SOIL PROFILE

DATA CARD   'PC-STABL3D Version 2.0'

NOTE: The previous data card is intended for identification of the PC-STABL3D data files. Therefore it must be printed exactly as it appears in the text of this manual, starting from the first left column.

DATA CARD   Number of cross sections in the X
direction

Number of cross sections in the Y
direction

DATA CARD   Coordinates X of cross sections
parallel to the Y axis

DATA CARD   Coordinates Y of cross sections
parallel to the X axis

NOTE: For the two previous data cards, put as many data entries as fit on each line, until the last value is reached.

DATA CARD   Coordinates Z for all nodes of the
cross section X_1 , starting from
the intersection of cross sections

X_1 and Y_1 , and for all subsequent intersections of X_1 and Y_i

NOTE: Repeat the previous data card for each consequent cross section X_i . Put as many data entries as fit on each line, until the last value is reached.

GENERATION LIMITS

DATA CARD   The number i of the cross section X_i corresponding to main sliding axis.

DATA CARD   Minimum coordinate $Y_{i \min}$ of the trial surfaces initiation region for the main sliding axis

Interval of trial surfaces initiation $\Delta Y_{i \min}$

Minimum coordinate $Y_{f \min}$ of the trial surfaces termination region for the main sliding axis

Interval of trial surface termination $\Delta Y_{f \min}$

DATA CARD   Minimum coordinate $X_{i \min}$ of the trial surfaces initiation region for the second cross section parallel to the Y axis (Y_2)

Interval of trial surface initiation $\Delta X_{i\min}$ for the same cross section

NOTE: Repeat previous data card for all consecutive cross sections parallel to the Y axis starting from cross section Y_3 , and terminating with the one before the last.

DATA CARD Minimum coordinate $Z_{\min}$, below which surfaces will not be generated for the each node of the first cross section parallel to the Y axis (Y_1)

Interval $\Delta Z_{\min}$, such that surfaces would not be generated above the coordinate $Z_{\min} + \Delta Z_{\min}$ for the corresponding node.

NOTE: For the previous data card put as many data values for the all consecutive nodes of the first cross section, then repeat this data card for consecutive cross sections parallel to the Y axis terminating with the last cross section.

SOIL PROPERTIES

DATA CARD Moist or saturated unit weight γ .

Strength intercept c , or c' for effective stress analysis.

Strength angle ϕ , or ϕ' for
effective stress analysis

NOTE: Repeat previous data card for each soil

WATER CONDITIONS

DATA CARD Water conditions representation
code. Enter 1 for pore pressure
ratio, 0 for piezometric surface

IF PORE PRESSURE RATIO IS SELECTED, THEN:

DATA CARD Pore pressure ratio r_u

IF PIEZOMETRIC SURFACE IS SELECTED:

DATA CARD Coordinate Z of the piezometric
surface for each node of the first
cross section parallel to the X
axis

*NOTE: Repeat previous data card for all consecutive cross section in
the Y direction, then repeat all soil condition data cards for
each soil layer*

5.0 GRAPHICAL PRE/POSTPROCESSOR

5.1 Overview

A graphical pre/postprocessor facilitates creation of the data files and allows the user to view geometry of the original slope and failure surfaces. The following operations can be performed using a pre/postprocessor:

- creating of the data file
- editing of the existing data file
- viewing the cross sections of the existing slope
- viewing the contour plan of the existing slope
- viewing the cross section of the slope after failure
- viewing the contour plan of the slope after failure

The convenience of the pre/postprocessor operation is facilitated by use of the sliding bar and pop-up menus. The picture of the main menu which appears on the computer screen after the program is invoked is shown on Figure 8. The user can choose one of the following options from the main menu:

1. Create Data File
2. Edit Existing Data File
3. Read Data File
4. Plot 3D Geometry
5. Plot Cross Sections
6. Run PC-STABL3D
7. Plot 3D Surface Contours
8. Plot Surface Cross Section

Figure 8   **Main Menu**

9. Exit Program

The user can examine the options, moving the highlighting bar with the "Up" and "Down" arrow keys, and choose the option by pressing the "Enter" key. The "Esc" key will bring the user to the previous screen or to the previous input field. The subsequent paragraphs describe a typical session with the pre/postprocessor.

5.2 Creation and Edition of Data Files

To start creation of the data file, choose "Create Data File" option and press "Enter". The program will prompt for the name of the file where the data are contained, as shown on Figure 9. Note that if a file with the same name already exists, it will be overwritten, and its contents will be lost. Therefore the user must be certain to use an original file name. An edition can be suspended at any time by pressing "Ctrl" and "End" keys simultaneously. After the name of the data file has been entered, press "Enter" to go to the next screen.

This screen (Figure 10) prompts for the number of grid points in X and Y directions and number of soil layers. As previously mentioned, the grid will be used to represent the soil profile. If an error has been made entering the numerical values, go back by pressing the "Esc" key. Press the "Enter" key after each value is entered to go to the next field. The next screen (Figure 11) prompts for the actual coordinates of the points. The user enters coordinates subsequently first for all points in X direction and then for all points in Y direction. After the grid has been entered, the computer prompts for the Z coordinate of the soil for each node of the mesh (Figure 12). This procedure

Figure 9 Prompt for Name of Data File

The image shows a computer screen with a dark background. A white rectangular box in the center contains the following text:

You will be entering the grid of X and Y points next.
This grid will be used to represent the soil profile.

Please enter: Number of points in X direction: 10
Number of points in Y direction: 10
Maximum number of Soils: 2

(Remember that the grid is in the Y direction)

Below this box, the text "Press CNTRL-END to suspend edition" is displayed. At the bottom of the screen, a white box contains the text "PC-STABL3D - 3 Dimensional Slope Stability Program" and "Version 2.2".

Figure 10 Prompt for the Number of Grid Points

Now you will enter the actual coordinates of the
point

For point X ( ), enter its coordinate value: x = 9

For point Y ( ), enter its coordinate value: y = 9

Remember to use consistent length units...

Press CTRL-END to suspend edition

HL STABL3D - 3 Dimensional Slope Stability Program
Version 2.2

Figure 11 Prompt for the Horizontal Coordinates of Nodes

Figure 12   Prompt for the Vertical Coordinates of Nodes

is then reiterated for each soil layer.

The following step is the entering of the failure surface generation constraints. As noted before, a principal axis of sliding should be defined first. Figure 13 shows the window for entering the above data. As can be seen, after the main sliding axis is defined, the computer prompts for the initial and termination values of Y for the main axis, and corresponding initiation and termination intervals. The next window (Figure 14) prompts for initiation values for surface generation in the direction parallel to the Y axis for each subsequent Y axis. These values are entered along with the corresponding surface generation intervals. The last part of the definition of failure surface generation is entering constraints in the vertical direction. As it was noted above, these constraints play a key role in the surface generation and therefore must be defined precisely.

After the failure surface generation limits are entered, the program prompts for the soil properties of the each soil layer (Figure 15). The user must enter unit weight of the soil, strength intercept, and strength shear angle. As mentioned above, the calculations in the program are performed using either total or effective stress approach, hence the appropriate strength parameters must be used for the input. The program does not require usage of any particular set of units; however, all units must be compatible.

The last part of the input procedure is the entering of the ground water data. Figure 16 shows the program prompt for the selection of the pore pressure model. The user will need enter "1" if the pore pressure ratio is

Data for Surface Generation:

Enter number N where X(N) would be the Principal Axis
of Sliding (ex: x(3)-> N=3) **5**

On the principal axis, enter search limit

Initiation value of Ymin **10**

Range of Ymin **3**

Termination value of Ymax **70**

Range of Ymax **3**

Press CNTRL-END to suspend edition

PC-STABL3D - 3 Dimensional Slope Stability Program
Version 2.2

Figure 13 Prompt for the Failure Surface Generation Data

Now you will enter the searching constrains for the
search in the Z (vertical) direction for all the grid:

For point X(   ), and Y(   ) enter value of Z_{min} = 20
delta Z = 4_

Remember to use consistent length units...

Press CNTRL-END to suspend edition

THBL3D - 3 Dimensional Slope Stability Program
Version 2.2

Figure 14   Prompt for the Initiation Values of Surface Generation

The screenshot shows a terminal window with a black background and white text. The main window has a thick black border. Inside, the text is as follows:

Soil Properties Data:

Now you will enter the strength properties of
the soils:

For soil number   , enter:

Unit weight 120
Cohesion 0
Shear Angle 28

!!! REMEMBER TO USE COMPATIBLE UNITS !!!

Press CTRL-END to suspend edition

At the bottom, a separate box contains the program title and version:

PC STABL3D - 3 Dimensional Slope Stability Program
Version 2.2

Figure 15   Prompt for the Soil Properties

PURE PRESSURE DATA:

Please select the porepressure model type from the list below:

- 1 - Pore pressure ratio (R_u)
- 2 - Piezometric Surfaces

For soil number , the model you choose is: **1**
(Enter 1 or 2)

Then enter R_u for this soil: **.10**

Press CNTRL-END to suspend edition

PC-STABL3D - 3 Dimensional Slope Stability Program
Version 2.2

Figure 16 Prompt for the Pore Pressure Model

chosen or "2" if the piezometric surfaces are chosen¹. If the pore pressure ratio is chosen, the prompt for the value of this ratio will appear, as shown on Figure 16. After this value is entered, the program prompts for the pore pressure model for the following soil layer. If the piezometric surfaces are chosen, the new window will appear (Figure 17) where the user will be asked for the "Z coordinate of the piezometric surface for each node of the defined coordinate mesh. The entering of the ground water data completes creation of the data file.

If errors in the problem geometry are encountered, the user can examine and edit the existing data file using the main menu option "Edit Existing Data File". The operations used to edit the existing data file are the same as those used to create it. The only difference is that the values of parameters will appear in the input fields instead of blanks (or zeros).

When the user finishes creating of the data files, the calculations using PC-STABL3D can be performed. In order to start PC-STABL3D program, the user must highlight the "Run PC-STABL3D" option from the main menu and press "Enter". The pre/postprocessor unloads itself from the computer's memory and loads PC-STABL3D instead. The program will prompt the user for the name of the input data file, output data file, the number of surfaces to be generated and, after the calculations are completed, for the name of the file where the critical surface data are to be written. After the program stops, invoke the pre/postprocessor at the DOS prompt to examine the geometry of the most critical surface.

---

¹ To input a number in the one-character input field, user will need first to delete existing character

---: PRESSURE DATA - Piezometric Line Data

For soil number   Please enter the piezometric value
for the coordinates below:

X(   ) =
Y(   ) =

--- piezometric value = 30

*** REMEMBER TO USE COMPATIBLE UNITS ***

CTRL END to suspend edition

PL STABL3D - 3 Dimensional Slope Stability Program
Version 2.2

Figure 17   Prompt for the Coordinates of Piezometric Surfaces

5.3 Examining of the Surface Geometry

After the data file is created, the next step in the input data preparation is to check a geometry of the created 3D surface. The pre/postprocessor provides two facilities for representation of the problem geometry: cross section plots, and contour plots. In order to perform plotting functions the input file must be read in first. This can be accomplished by the choosing of the "Read Data File" option from the main menu. The following window with a pop-up file listing (Figure 18) will prompt the user for the name of the desired data file. The user can examine the directory listing in the pop-up window, using "Up" and "Down" arrow keys, or the "Home" key to go to the beginning of the list and the "End" key to go to the end of the list. After the sliding bar is over the desired file, choose it by pressing the "Enter" key. The next window appears (Figure 19) with the prompt for the cross-section number to be plotted, where the user can choose to plot any of the cross section corresponding to the defined X axis. An example of a cross sections is shown on the Figure 20. Besides with the ground surface and soil boundary lines, the program plots surface generation limits (red on the color monitor), which can be used to check the geometry of the problem. It is possible to dump the image on the screen to the printer, if the screen dump program supports EGA or VGA graphics modes.

A similar procedure leads to the plotting of contour lines. The user must first read in the data file and choose the "Plot 3D Geometry" option from the main menu. The program will then ask for the number of the soil layer

Figure 18 Pop-up File Listing

Figure 19 Prompt for the Cross Section Number

Figure 20 Example of a Cross Section

to be plotted and the number of the contour levels to use. In order to use this option a color monitor is required because the contour lines differ in color, and it is impossible to distinguish among them on the black and white or monochrome screen.

The operations to preview the critical surface geometry are essentially the same as these to preview geometry of the initial profile. To preview the cross sections of the critical surface, the user will need to choose option "Plot 3D Surface Contours" from the main menu. From the pop-up directory window which will appear on the screen, select the file containing critical surface data (this file was generated by PC-STABL3D). The program will read the selected file and prompt for the number of contour levels to be plotted. Input the desired number of contour levels and press "Enter". The plot will appear on the screen (Figure 21). To return to the main menu press any key.

The pre/postprocessor allows plotting cross sections of the most critical failure surface. In order to perform this operation the user must choose the option "Plot Surface Cross Section" from the main menu and press "Enter". The pop-up window with the directory listing will appear, and the user will choose the desired file with the most critical surface geometry and press "Enter". The program then will ask for the cross section number to be plotted. After the required number is entered, the plot with the cross section of the most critical failure surface will appear on the screen.

To exit pre/postprocessor user must highlight the "Exit Program" option from the main menu.

Figure 21 Example of a Contour Plot

6.0 MISCELLANEOUS INFORMATION

6.1 Modification and Revisions of PC-STABL3D

The first version of the program PC-STABL3D and pre/postprocessor was developed in 1986 (Thomaz, 1986). Since that time the program has been used mostly for educational purposes.

The current modification of this program was made by G. Moshkovich. The following modifications were implemented:

1. Data files can be not just created, but also, edited using the enhanced menus of the pre/postprocessor.
2. During the creation and edition of the data files the user can now move forward and backward in the menus while the previous version allowed forward motion only.
3. The system of error messages was included in the preprocessor to enhance the error tracking capabilities of the program.
4. The surface generation routines for the main program were optimized in order to increase their speed and facilitate further modifications.

6.2 Units

All units used for any one problem must be consistent. The printed output is limited to the following units for dimensioning.

Length	(FT)
Unit Weight	(PCF)
Stress	(PSF)
Angles	(DEG)

Metric units or any set of consistent units can be used. It must be kept in mind, however, that the printed output will bear the units listed above. A consistent set of metric units for example would be:

Length	(M)
Unit Weight	(KG/M ³)
Stress	(KG/M ²)
Angles	(DEG)

6.3 Problem Size Limitations

The limitations of the program are defined by the size of the arrays. The arrays of PC-STABL3D were assigned the following dimensions:

Data	Maximum Number ¹
Number of points in X direction	15
Number of points in Y direction	15
Number of soils	5

7.0 Results

The current version of the PC-STABL3D was modified in order to improve interactive interface of the program and enhance mathematical routines for generation of the random surfaces.

The modifications made include:

1. Data files can be now not just created, but also, edited using the enhanced menus of the pre/postprocessor.
2. During the creation and edition of the data files the user can now move forward and backward in the menus while the previous version allowed forward motion only.
3. The system of error messages was included in the preprocessor to enhance the error tracking capabilities of the program.

---

¹ The program can be adjusted to handle larger problems by changing dimension statements. The availability of the computer's memory core will take precedence with regard to how large a problem can be ultimately handled.

4. The surface generation routines for the main program were optimized in order to increase their speed and facilitate further modifications.
5. The program performs statistical evaluation of the calculated minimum factor of safety using the Weibull's distribution, and estimates value of minimum statistical factor of safety.

8.0 Conclusions

Based on the results of the research, the following conclusions were made regarding the applicability of the PC-STABL3D to the solution of general three-dimensional slope stability problems.

1. The program can perform three-dimensional stability analysis of slopes if the number of sections along or across the given slope does not exceed 10.
2. The maximum number of soil layers allowed in the program is 5. This and the above limitations are conditions of the availability of the computer memory required for the program.
3. The program performs calculations using the total weight and total boundary force approach. However if the pore pressure is measured or predicted, either a total or an effective stress analysis may be undertaken.
4. The factor of safety calculated using the three-dimensional analysis is generally higher than the one calculated using the planar stress analysis.

REFERENCES

- Benjamin, J. R. and Cornell, C. A. (1970), "Probability, Statistics and Decision for Civil Engineers", McGraw Hill, New York
- Chen, R. H. (1981) "Three Dimensional Slope Stability Analysis", Report JHRP-81-17, Purdue University, West Lafayette, Indiana.
- Meyer, S. L.(1975) "Data Analysis for Scientists and Engineers", John Wiley & Sons, New York
- Thomaz J.E. (1986) "A General Method for Three Dimensional Slope Stability Analysis", Report JHRP 86-12, Purdue University, West Lafayette, Indiana.

APPENDICES

APPENDIX A   STATISTICAL ANALYSIS

The program provides statistical estimation of the reliability of critical factor of safety. The estimation is needed since the factors of safety generated by the program are based on the analysis of random surfaces. In order to perform this statistical evaluation, the distribution function for the distribution of resulting factors of safety has been defined.

The distribution function for factors of safety satisfies the following conditions:

- It is limited to the left, since the minimum value of the factor of safety is of interest.
- It is open to the right since the maximum factor of safety is indeterminate
- Function, first and second derivatives of the function must be continuous.

The distribution which corresponds to the above conditions is termed extreme value distribution, or also the Weibull's distribution. The probability density function and cumulative distribution function for this distribution are:

$$f_z = \frac{k}{u-\epsilon} \left(\frac{z-\epsilon}{u-\epsilon} \right)^{k-1} \exp \left[- \left(\frac{z-\epsilon}{u-\epsilon} \right)^k \right] \quad z \geq \epsilon \quad (\text{A1})$$

$$F_Z(z) = 1 - \exp\left[-\left(\frac{z-\epsilon}{u-\epsilon}\right)^k\right] \quad z \geq \epsilon \quad (\text{A2})$$

The moments of this distribution are:

$$m_Z = \epsilon + (u-\epsilon)\Gamma\left(1+\frac{1}{k}\right) \quad (\text{A3})$$

$$\sigma_Z^2 = (u-\epsilon)^2 \left[\Gamma\left(1+\frac{2}{k}\right) - \Gamma^2\left(1+\frac{1}{k}\right) \right] \quad (\text{A4})$$

$$E[(Z-\epsilon)^l] = (u-\epsilon)^l \Gamma\left(1+\frac{l}{k}\right) \quad (\text{A5})$$

where

$$\Gamma(k) = \int_0^{\infty} e^{-u} u^{k-1} du \quad (\text{A6})$$

is called the Gamma function.

The first and second moments of the calculated factors of safety can be estimated and, hence, parameters of the distribution can be defined. The program calculates parameter ϵ , which is the left boundary of the distribution or minimum statistical factor of safety. Thus, the user can evaluate how close

the actual minimum factor of safety to the one computed from the Weibull's distribution.

The Weibull's distribution can not evaluate a range of factors of safety until the minimum statistical factor of safety is computed. This is why the statistical factor of safety can have any value, even a negative one. In this case the program will warn the user and suggest that either: the limiting generation boundaries be redefined or the program be run with a larger number of iterations.

The user can use a statistical estimation of the minimum factor of safety to evaluate reliability of the calculated factor of safety. If the statistically estimated factor of safety is considerably less (10%-15%) than the calculated minimum factor of safety, than the variability of the generated surfaces is very high. The best way to decrease variability within the safety factors is to narrow the limits of surface generation. The user may attempt to change both the vertical and horizontal limits of surface generation in order to achieve plausible value for the statistical estimation of minimum factor of safety. The procedure for the search of the most critical failure surface is, of course, trial and error method, and therefore the user's engineering judgment and experience are important factors.

The statistically estimated minimum factor of safety can also help the user to evaluate the reliability of the computed minimum factor of safety for the cases of difficult soil conditions. Such complicating soil conditions can be, for example, weak layers. If the limits of surface generation are assigned such that there is a possibility for the failure surface to pass either through the

stronger soil or through a weak layer, the scatter of the resulting factors of safety will be high. Therefore the difference between the statistically estimated and the computed minimum factors of safety may be considerable. In order to decrease the above difference the user can shift the vertical limits of surface generation up or down to include as much of the weak layer as possible in the generation limits.

The user, however, should exercise caution using the statistically determined minimum factor of safety, since its value can be misleading if the number of generated surfaces is low, and the generation limits are wide.

APPENDIX B   EXAMPLES**Example 1**

Geometry of the Example 1 is shown on Figure B1. The following soil parameters were used:

Unit weight of soil   100 pcf
Cohesion   500 psf
Friction angle   0 deg

Figure B1   Illustration for Example 1

15	13
1	
40	60 80 100 120 130
0	20 40 60 80 100 120
100	100 100 100 100 100
100	100 100 100 100 100
100	100 100 100 100 100
100	100 100 100 100 100
100	100 100 100 100 103.3
100	100 100 100 103.33 11
100	100 100 100 106.67 11
100	100 100 100 106.67 11
100	100 100 100 106.67 11
100	100 100 100 106.67 11
100	100 100 100 106.67 11
100	100 100 100 103.33 11
100	100 100 100 100 103.3
100	100 100 100 100 100
100	100 100 100 100 100
100	100 100 100 100 100
8	
20	40 110 60
120	40 200 40
120	40 200 40
120	40 200 40
100	40 220 40
100	40 220 40
100	40 220 40
100	40 220 40
100	40 220 40
120	40 200 40
120	40 200 40
120	40 200 40
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
79	20 79 20 79 20 79
100	500 0
1	
0	

OUTPUT FILE

```
*****
*           PC-STABL3D           *
* 3-DIMENSIOANL SLOPE STABILITY PROGRAM *
* FROM PURDUE UNIVERSITY      *
* VERSION 2.1                  *
*****
```

Number of X coordinates : 15
 Number of Y coordinates : 13
 Number of soils : 1

X COORDINATES :

```
coordinate X( 1) = 40.00
coordinate X( 2) = 60.00
coordinate X( 3) = 80.00
coordinate X( 4) = 100.00
coordinate X( 5) = 120.00
coordinate X( 6) = 130.00
coordinate X( 7) = 160.00
coordinate X( 8) = 180.00
coordinate X( 9) = 200.00
coordinate X(10) = 230.00
coordinate X(11) = 240.00
coordinate X(12) = 260.00
coordinate X(13) = 280.00
coordinate X(14) = 300.00
coordinate X(15) = 320.00
```

Y COORDINATES :

```
coordinate Y( 1) = .00
coordinate Y( 2) = 20.00
coordinate Y( 3) = 40.00
coordinate Y( 4) = 60.00
coordinate Y( 5) = 80.00
coordinate Y( 6) = 100.00
coordinate Y( 7) = 120.00
coordinate Y( 8) = 130.00
coordinate Y( 9) = 140.00
coordinate Y(10) = 160.00
coordinate Y(11) = 180.00
coordinate Y(12) = 200.00
coordinate Y(13) = 230.00
```

COORDINATES Z FOR SOIL # 1

```
Z ( 1, 1) = 100.00  Z ( 1, 2) = 100.00  Z ( 1, 3) = 100.00
Z ( 1, 4) = 100.00  Z ( 1, 5) = 100.00  Z ( 1, 6) = 100.00
Z ( 1, 7) = 100.00  Z ( 1, 8) = 100.00  Z ( 1, 9) = 100.00
Z ( 1,10) = 100.00  Z ( 1,11) = 100.00  Z ( 1,12) = 100.00
Z ( 1,13) = 100.00  Z ( 2, 1) = 100.00  Z ( 2, 2) = 100.00
Z ( 2, 3) = 100.00  Z ( 2, 4) = 100.00  Z ( 2, 5) = 100.00
Z ( 2, 6) = 100.00  Z ( 2, 7) = 103.33  Z ( 2, 8) = 106.67
Z ( 2, 9) = 106.67  Z ( 2,10) = 106.67  Z ( 2,11) = 106.67
```

Z (2,12) = 106.67	Z (2,13) = 106.67	Z (3, 1) = 100.00
Z (3, 2) = 100.00	Z (3, 3) = 100.00	Z (3, 4) = 100.00
Z (3, 5) = 100.00	Z (3, 6) = 103.33	Z (3, 7) = 110.00
Z (3, 8) = 113.33	Z (3, 9) = 113.33	Z (3,10) = 113.33
Z (3,11) = 113.33	Z (3,12) = 113.33	Z (3,13) = 113.33
Z (4, 1) = 100.00	Z (4, 2) = 100.00	Z (4, 3) = 100.00
Z (4, 4) = 100.00	Z (4, 5) = 103.33	Z (4, 6) = 110.00
Z (4, 7) = 116.67	Z (4, 8) = 120.00	Z (4, 9) = 120.00
Z (4,10) = 120.00	Z (4,11) = 120.00	Z (4,12) = 120.00
Z (4,13) = 120.00	Z (5, 1) = 100.00	Z (5, 2) = 100.00
Z (5, 3) = 100.00	Z (5, 4) = 103.33	Z (5, 5) = 110.00
Z (5, 6) = 116.67	Z (5, 7) = 123.33	Z (5, 8) = 126.67
Z (5, 9) = 126.67	Z (5,10) = 126.67	Z (5,11) = 126.67
Z (5,12) = 126.67	Z (5,13) = 126.67	Z (6, 1) = 100.00
Z (6, 2) = 100.00	Z (6, 3) = 100.00	Z (6, 4) = 106.67
Z (6, 5) = 113.33	Z (6, 6) = 120.00	Z (6, 7) = 126.67
Z (6, 8) = 130.00	Z (6, 9) = 130.00	Z (6,10) = 130.00
Z (6,11) = 130.00	Z (6,12) = 130.00	Z (6,13) = 130.00
Z (7, 1) = 100.00	Z (7, 2) = 100.00	Z (7, 3) = 100.00
Z (7, 4) = 106.67	Z (7, 5) = 113.33	Z (7, 6) = 120.00
Z (7, 7) = 126.67	Z (7, 8) = 130.00	Z (7, 9) = 130.00
Z (7,10) = 130.00	Z (7,11) = 130.00	Z (7,12) = 130.00
Z (7,13) = 130.00	Z (8, 1) = 100.00	Z (8, 2) = 100.00
Z (8, 3) = 100.00	Z (8, 4) = 106.67	Z (8, 5) = 113.33
Z (8, 6) = 120.00	Z (8, 7) = 126.67	Z (8, 8) = 130.00
Z (8, 9) = 130.00	Z (8,10) = 130.00	Z (8,11) = 130.00
Z (8,12) = 130.00	Z (8,13) = 130.00	Z (9, 1) = 100.00
Z (9, 2) = 100.00	Z (9, 3) = 100.00	Z (9, 4) = 106.67
Z (9, 5) = 113.33	Z (9, 6) = 120.00	Z (9, 7) = 126.67
Z (9, 8) = 130.00	Z (9, 9) = 130.00	Z (9,10) = 130.00
Z (9,11) = 130.00	Z (9,12) = 130.00	Z (9,13) = 130.00
Z (10, 1) = 100.00	Z (10, 2) = 100.00	Z (10, 3) = 100.00
Z (10, 4) = 106.67	Z (10, 5) = 113.33	Z (10, 6) = 120.00
Z (10, 7) = 126.67	Z (10, 8) = 130.00	Z (10, 9) = 130.00
Z (10,10) = 130.00	Z (10,11) = 130.00	Z (10,12) = 130.00
Z (10,13) = 130.00	Z (11, 1) = 100.00	Z (11, 2) = 100.00
Z (11, 3) = 100.00	Z (11, 4) = 103.33	Z (11, 5) = 110.00
Z (11, 6) = 116.67	Z (11, 7) = 123.33	Z (11, 8) = 126.67
Z (11, 9) = 126.67	Z (11,10) = 126.67	Z (11,11) = 126.67
Z (11,12) = 126.67	Z (11,13) = 126.67	Z (12, 1) = 100.00
Z (12, 2) = 100.00	Z (12, 3) = 100.00	Z (12, 4) = 100.00
Z (12, 5) = 103.33	Z (12, 6) = 110.00	Z (12, 7) = 116.67
Z (12, 8) = 120.00	Z (12, 9) = 120.00	Z (12,10) = 120.00
Z (12,11) = 120.00	Z (12,12) = 120.00	Z (12,13) = 120.00
Z (13, 1) = 100.00	Z (13, 2) = 100.00	Z (13, 3) = 100.00
Z (13, 4) = 100.00	Z (13, 5) = 100.00	Z (13, 6) = 103.33
Z (13, 7) = 110.00	Z (13, 8) = 113.33	Z (13, 9) = 113.33
Z (13,10) = 113.33	Z (13,11) = 113.33	Z (13,12) = 113.33
Z (13,13) = 113.33	Z (14, 1) = 100.00	Z (14, 2) = 100.00
Z (14, 3) = 100.00	Z (14, 4) = 100.00	Z (14, 5) = 100.00
Z (14, 6) = 100.00	Z (14, 7) = 103.33	Z (14, 8) = 106.67
Z (14, 9) = 106.67	Z (14,10) = 106.67	Z (14,11) = 106.67
Z (14,12) = 106.67	Z (14,13) = 106.67	Z (15, 1) = 100.00
Z (15, 2) = 100.00	Z (15, 3) = 100.00	Z (15, 4) = 100.00
Z (15, 5) = 100.00	Z (15, 6) = 100.00	Z (15, 7) = 100.00
Z (15, 8) = 100.00	Z (15, 9) = 100.00	Z (15,10) = 100.00
Z (15,11) = 100.00	Z (15,12) = 100.00	Z (15,13) = 100.00

BOUNDARIES FOR GENERATION IN THE Y DIRECTION :

Main Axis of Sliding was established at X = 180.00

Minimum point for initiation of polygons at Y = 20.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at Y = 110.00
 Interval for termination of polygons is = 60.00

BOUNDARIES FOR GENERATION IN THE X DIRECTION :

Coordinate Y = .00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 20.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 40.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 60.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 80.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 100.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 120.00
 Minimum point for initiation of polygons at X = 100.00

Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 130.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 140.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 160.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 180.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 200.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 230.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

BOUNDARIES FOR GENERATION IN THE Z DIRECTION :

Zmin(1, 1) = 79.00	Delta Z(1, 1) = 20.00
Zmin(1, 2) = 79.00	Delta Z(1, 2) = 20.00
Zmin(1, 3) = 79.00	Delta Z(1, 3) = 20.00
Zmin(1, 4) = 79.00	Delta Z(1, 4) = 20.00
Zmin(1, 5) = 79.00	Delta Z(1, 5) = 20.00
Zmin(1, 6) = 79.00	Delta Z(1, 6) = 20.00
Zmin(1, 7) = 79.00	Delta Z(1, 7) = 20.00
Zmin(1, 8) = 79.00	Delta Z(1, 8) = 20.00
Zmin(1, 9) = 79.00	Delta Z(1, 9) = 20.00

Zmin(1,10) =	79.00	Delta Z(1,10) =	20.00
Zmin(1,11) =	79.00	Delta Z(1,11) =	20.00
Zmin(1,12) =	79.00	Delta Z(1,12) =	20.00
Zmin(1,13) =	79.00	Delta Z(1,13) =	20.00
Zmin(2, 1) =	79.00	Delta Z(2, 1) =	20.00
Zmin(2, 2) =	79.00	Delta Z(2, 2) =	20.00
Zmin(2, 3) =	79.00	Delta Z(2, 3) =	20.00
Zmin(2, 4) =	79.00	Delta Z(2, 4) =	20.00
Zmin(2, 5) =	79.00	Delta Z(2, 5) =	20.00
Zmin(2, 6) =	79.00	Delta Z(2, 6) =	20.00
Zmin(2, 7) =	79.00	Delta Z(2, 7) =	21.00
Zmin(2, 8) =	79.00	Delta Z(2, 8) =	24.00
Zmin(2, 9) =	79.00	Delta Z(2, 9) =	24.00
Zmin(2,10) =	79.00	Delta Z(2,10) =	24.00
Zmin(2,11) =	79.00	Delta Z(2,11) =	24.00
Zmin(2,12) =	79.00	Delta Z(2,12) =	24.00
Zmin(2,13) =	79.00	Delta Z(2,13) =	24.00
Zmin(3, 1) =	79.00	Delta Z(3, 1) =	20.00
Zmin(3, 2) =	79.00	Delta Z(3, 2) =	20.00
Zmin(3, 3) =	79.00	Delta Z(3, 3) =	20.00
Zmin(3, 4) =	79.00	Delta Z(3, 4) =	20.00
Zmin(3, 5) =	79.00	Delta Z(3, 5) =	20.00
Zmin(3, 6) =	79.00	Delta Z(3, 6) =	22.00
Zmin(3, 7) =	79.00	Delta Z(3, 7) =	25.00
Zmin(3, 8) =	79.00	Delta Z(3, 8) =	30.00
Zmin(3, 9) =	79.00	Delta Z(3, 9) =	30.00
Zmin(3,10) =	79.00	Delta Z(3,10) =	30.00
Zmin(3,11) =	79.00	Delta Z(3,11) =	30.00
Zmin(3,12) =	79.00	Delta Z(3,12) =	30.00
Zmin(3,13) =	79.00	Delta Z(3,13) =	30.00
Zmin(4, 1) =	79.00	Delta Z(4, 1) =	20.00
Zmin(4, 2) =	79.00	Delta Z(4, 2) =	20.00
Zmin(4, 3) =	79.00	Delta Z(4, 3) =	20.00
Zmin(4, 4) =	79.00	Delta Z(4, 4) =	20.00
Zmin(4, 5) =	79.00	Delta Z(4, 5) =	22.00
Zmin(4, 6) =	79.00	Delta Z(4, 6) =	25.00
Zmin(4, 7) =	79.00	Delta Z(4, 7) =	32.00
Zmin(4, 8) =	79.00	Delta Z(4, 8) =	35.00
Zmin(4, 9) =	79.00	Delta Z(4, 9) =	35.00
Zmin(4,10) =	79.00	Delta Z(4,10) =	35.00
Zmin(4,11) =	79.00	Delta Z(4,11) =	35.00
Zmin(4,12) =	79.00	Delta Z(4,12) =	35.00
Zmin(4,13) =	79.00	Delta Z(4,13) =	35.00
Zmin(5, 1) =	79.00	Delta Z(5, 1) =	20.00
Zmin(5, 2) =	79.00	Delta Z(5, 2) =	20.00
Zmin(5, 3) =	79.00	Delta Z(5, 3) =	20.00
Zmin(5, 4) =	79.00	Delta Z(5, 4) =	22.00
Zmin(5, 5) =	79.00	Delta Z(5, 5) =	25.00
Zmin(5, 6) =	79.00	Delta Z(5, 6) =	30.00
Zmin(5, 7) =	79.00	Delta Z(5, 7) =	35.00
Zmin(5, 8) =	79.00	Delta Z(5, 8) =	35.00
Zmin(5, 9) =	79.00	Delta Z(5, 9) =	35.00
Zmin(5,10) =	79.00	Delta Z(5,10) =	35.00
Zmin(5,11) =	79.00	Delta Z(5,11) =	35.00
Zmin(5,12) =	79.00	Delta Z(5,12) =	35.00
Zmin(5,13) =	79.00	Delta Z(5,13) =	35.00
Zmin(6, 1) =	79.00	Delta Z(6, 1) =	20.00
Zmin(6, 2) =	79.00	Delta Z(6, 2) =	20.00

Zmin(6, 3) =	79.00	Delta Z(6, 3) =	20.00
Zmin(6, 4) =	79.00	Delta Z(6, 4) =	25.00
Zmin(6, 5) =	79.00	Delta Z(6, 5) =	30.00
Zmin(6, 6) =	79.00	Delta Z(6, 6) =	35.00
Zmin(6, 7) =	79.00	Delta Z(6, 7) =	40.00
Zmin(6, 8) =	79.00	Delta Z(6, 8) =	45.00
Zmin(6, 9) =	79.00	Delta Z(6, 9) =	45.00
Zmin(6,10) =	79.00	Delta Z(6,10) =	45.00
Zmin(6,11) =	79.00	Delta Z(6,11) =	45.00
Zmin(6,12) =	79.00	Delta Z(6,12) =	45.00
Zmin(6,13) =	79.00	Delta Z(6,13) =	45.00
Zmin(7, 1) =	79.00	Delta Z(7, 1) =	20.00
Zmin(7, 2) =	79.00	Delta Z(7, 2) =	20.00
Zmin(7, 3) =	79.00	Delta Z(7, 3) =	20.00
Zmin(7, 4) =	79.00	Delta Z(7, 4) =	25.00
Zmin(7, 5) =	79.00	Delta Z(7, 5) =	30.00
Zmin(7, 6) =	79.00	Delta Z(7, 6) =	35.00
Zmin(7, 7) =	79.00	Delta Z(7, 7) =	40.00
Zmin(7, 8) =	79.00	Delta Z(7, 8) =	45.00
Zmin(7, 9) =	79.00	Delta Z(7, 9) =	45.00
Zmin(7,10) =	79.00	Delta Z(7,10) =	45.00
Zmin(7,11) =	79.00	Delta Z(7,11) =	45.00
Zmin(7,12) =	79.00	Delta Z(7,12) =	45.00
Zmin(7,13) =	79.00	Delta Z(7,13) =	45.00
Zmin(8, 1) =	79.00	Delta Z(8, 1) =	20.00
Zmin(8, 2) =	79.00	Delta Z(8, 2) =	20.00
Zmin(8, 3) =	79.00	Delta Z(8, 3) =	20.00
Zmin(8, 4) =	79.00	Delta Z(8, 4) =	25.00
Zmin(8, 5) =	79.00	Delta Z(8, 5) =	30.00
Zmin(8, 6) =	79.00	Delta Z(8, 6) =	35.00
Zmin(8, 7) =	79.00	Delta Z(8, 7) =	40.00
Zmin(8, 8) =	79.00	Delta Z(8, 8) =	45.00
Zmin(8, 9) =	79.00	Delta Z(8, 9) =	45.00
Zmin(8,10) =	79.00	Delta Z(8,10) =	45.00
Zmin(8,11) =	79.00	Delta Z(8,11) =	45.00
Zmin(8,12) =	79.00	Delta Z(8,12) =	45.00
Zmin(8,13) =	79.00	Delta Z(8,13) =	45.00
Zmin(9, 1) =	79.00	Delta Z(9, 1) =	20.00
Zmin(9, 2) =	79.00	Delta Z(9, 2) =	20.00
Zmin(9, 3) =	79.00	Delta Z(9, 3) =	20.00
Zmin(9, 4) =	79.00	Delta Z(9, 4) =	25.00
Zmin(9, 5) =	79.00	Delta Z(9, 5) =	30.00
Zmin(9, 6) =	79.00	Delta Z(9, 6) =	35.00
Zmin(9, 7) =	79.00	Delta Z(9, 7) =	40.00
Zmin(9, 8) =	79.00	Delta Z(9, 8) =	45.00
Zmin(9, 9) =	79.00	Delta Z(9, 9) =	45.00
Zmin(9,10) =	79.00	Delta Z(9,10) =	45.00
Zmin(9,11) =	79.00	Delta Z(9,11) =	45.00
Zmin(9,12) =	79.00	Delta Z(9,12) =	45.00
Zmin(9,13) =	79.00	Delta Z(9,13) =	45.00
Zmin(10, 1) =	79.00	Delta Z(10, 1) =	20.00
Zmin(10, 2) =	79.00	Delta Z(10, 2) =	20.00
Zmin(10, 3) =	79.00	Delta Z(10, 3) =	20.00
Zmin(10, 4) =	79.00	Delta Z(10, 4) =	25.00
Zmin(10, 5) =	79.00	Delta Z(10, 5) =	30.00
Zmin(10, 6) =	79.00	Delta Z(10, 6) =	35.00
Zmin(10, 7) =	79.00	Delta Z(10, 7) =	40.00
Zmin(10, 8) =	79.00	Delta Z(10, 8) =	45.00

Zmin(10, 9) =	79.00	Delta Z(10, 9) =	45.00
Zmin(10,10) =	79.00	Delta Z(10,10) =	45.00
Zmin(10,11) =	79.00	Delta Z(10,11) =	45.00
Zmin(10,12) =	79.00	Delta Z(10,12) =	45.00
Zmin(10,13) =	79.00	Delta Z(10,13) =	45.00
Zmin(11, 1) =	79.00	Delta Z(11, 1) =	20.00
Zmin(11, 2) =	79.00	Delta Z(11, 2) =	20.00
Zmin(11, 3) =	79.00	Delta Z(11, 3) =	20.00
Zmin(11, 4) =	79.00	Delta Z(11, 4) =	22.00
Zmin(11, 5) =	79.00	Delta Z(11, 5) =	25.00
Zmin(11, 6) =	79.00	Delta Z(11, 6) =	30.00
Zmin(11, 7) =	79.00	Delta Z(11, 7) =	35.00
Zmin(11, 8) =	79.00	Delta Z(11, 8) =	35.00
Zmin(11, 9) =	79.00	Delta Z(11, 9) =	35.00
Zmin(11,10) =	79.00	Delta Z(11,10) =	35.00
Zmin(11,11) =	79.00	Delta Z(11,11) =	35.00
Zmin(11,12) =	79.00	Delta Z(11,12) =	35.00
Zmin(11,13) =	79.00	Delta Z(11,13) =	35.00
Zmin(12, 1) =	79.00	Delta Z(12, 1) =	20.00
Zmin(12, 2) =	79.00	Delta Z(12, 2) =	20.00
Zmin(12, 3) =	79.00	Delta Z(12, 3) =	20.00
Zmin(12, 4) =	79.00	Delta Z(12, 4) =	20.00
Zmin(12, 5) =	79.00	Delta Z(12, 5) =	22.00
Zmin(12, 6) =	79.00	Delta Z(12, 6) =	25.00
Zmin(12, 7) =	79.00	Delta Z(12, 7) =	32.00
Zmin(12, 8) =	79.00	Delta Z(12, 8) =	35.00
Zmin(12, 9) =	79.00	Delta Z(12, 9) =	35.00
Zmin(12,10) =	79.00	Delta Z(12,10) =	35.00
Zmin(12,11) =	79.00	Delta Z(12,11) =	35.00
Zmin(12,12) =	79.00	Delta Z(12,12) =	35.00
Zmin(12,13) =	79.00	Delta Z(12,13) =	35.00
Zmin(13, 1) =	79.00	Delta Z(13, 1) =	20.00
Zmin(13, 2) =	79.00	Delta Z(13, 2) =	20.00
Zmin(13, 3) =	79.00	Delta Z(13, 3) =	20.00
Zmin(13, 4) =	79.00	Delta Z(13, 4) =	20.00
Zmin(13, 5) =	79.00	Delta Z(13, 5) =	20.00
Zmin(13, 6) =	79.00	Delta Z(13, 6) =	22.00
Zmin(13, 7) =	79.00	Delta Z(13, 7) =	25.00
Zmin(13, 8) =	79.00	Delta Z(13, 8) =	30.00
Zmin(13, 9) =	79.00	Delta Z(13, 9) =	30.00
Zmin(13,10) =	79.00	Delta Z(13,10) =	30.00
Zmin(13,11) =	79.00	Delta Z(13,11) =	30.00
Zmin(13,12) =	79.00	Delta Z(13,12) =	30.00
Zmin(13,13) =	79.00	Delta Z(13,13) =	30.00
Zmin(14, 1) =	79.00	Delta Z(14, 1) =	20.00
Zmin(14, 2) =	79.00	Delta Z(14, 2) =	20.00
Zmin(14, 3) =	79.00	Delta Z(14, 3) =	20.00
Zmin(14, 4) =	79.00	Delta Z(14, 4) =	20.00
Zmin(14, 5) =	79.00	Delta Z(14, 5) =	20.00
Zmin(14, 6) =	79.00	Delta Z(14, 6) =	20.00
Zmin(14, 7) =	79.00	Delta Z(14, 7) =	21.00
Zmin(14, 8) =	79.00	Delta Z(14, 8) =	24.00
Zmin(14, 9) =	79.00	Delta Z(14, 9) =	24.00
Zmin(14,10) =	79.00	Delta Z(14,10) =	24.00
Zmin(14,11) =	79.00	Delta Z(14,11) =	24.00
Zmin(14,12) =	79.00	Delta Z(14,12) =	24.00
Zmin(14,13) =	79.00	Delta Z(14,13) =	24.00
Zmin(15, 1) =	79.00	Delta Z(15, 1) =	20.00

Zmin(15, 2) = 79.00   Delta Z(15, 2) = 20.00
 Zmin(15, 3) = 79.00   Delta Z(15, 3) = 20.00
 Zmin(15, 4) = 79.00   Delta Z(15, 4) = 20.00
 Zmin(15, 5) = 79.00   Delta Z(15, 5) = 20.00
 Zmin(15, 6) = 79.00   Delta Z(15, 6) = 20.00
 Zmin(15, 7) = 79.00   Delta Z(15, 7) = 20.00
 Zmin(15, 8) = 79.00   Delta Z(15, 8) = 20.00
 Zmin(15, 9) = 79.00   Delta Z(15, 9) = 20.00
 Zmin(15,10) = 79.00   Delta Z(15,10) = 20.00
 Zmin(15,11) = 79.00   Delta Z(15,11) = 20.00
 Zmin(15,12) = 79.00   Delta Z(15,12) = 20.00
 Zmin(15,13) = 79.00   Delta Z(15,13) = 20.00

SOIL PROPERTIES :

Soil	Unit Weight	Cohesion	Shear Angle

1	100.00	500.0	.0

Ru of soil # 1 = .00

TEN MOST CRITICAL SURFACES :

SURFACE # 1

Coordinate X	Coordinate Y	Coordinate Z
180.000	41.947	.000
154.579	60.000	386.448
160.000	60.000	90.326
180.000	60.000	90.293
200.000	60.000	90.326
205.421	60.000	513.552
137.294	80.000	343.236
160.000	80.000	92.027
180.000	80.000	91.997
200.000	80.000	92.027
222.706	80.000	556.764
120.010	100.000	300.024
130.000	100.000	101.952
160.000	100.000	83.505
180.000	100.000	82.124
200.000	100.000	83.505
230.000	100.000	101.952
239.990	100.000	599.976
112.534	120.000	281.335
120.000	120.000	104.000
130.000	120.000	96.747
160.000	120.000	86.576
180.000	120.000	86.295
200.000	120.000	86.576
230.000	120.000	96.747
240.000	120.000	104.000
247.466	120.000	618.665

114.857	130.000	287.141
120.000	130.000	107.689
130.000	130.000	105.583
160.000	130.000	99.710
180.000	130.000	97.431
200.000	130.000	99.710
230.000	130.000	105.583
240.000	130.000	107.689
245.143	130.000	612.859
129.358	140.000	323.394
130.000	140.000	117.225
160.000	140.000	116.657
180.000	140.000	116.475
200.000	140.000	116.657
230.000	140.000	117.225
230.642	140.000	576.606
140.000	160.000	350.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	550.000
180.000	161.869	.000

Factor of Safety = 1.662

SURFACE # 2

Coordinate X	Coordinate Y	Coordinate Z
180.000	22.640	.000
137.565	40.000	343.912
160.000	40.000	85.030
180.000	40.000	83.057
200.000	40.000	85.030
222.435	40.000	556.088
124.614	60.000	311.536
130.000	60.000	89.673
160.000	60.000	81.507
180.000	60.000	79.000
200.000	60.000	81.507
230.000	60.000	89.673
235.386	60.000	588.464
111.664	80.000	279.159
120.000	80.000	101.000
130.000	80.000	104.000
160.000	80.000	94.851
180.000	80.000	91.949
200.000	80.000	94.851
230.000	80.000	104.000
240.000	80.000	101.000
248.336	80.000	620.841
109.278	100.000	273.194
120.000	100.000	104.000
130.000	100.000	109.000
160.000	100.000	109.000
180.000	100.000	109.000
200.000	100.000	109.000

230.000	100.000	109.000
240.000	100.000	104.000
250.722	100.000	626.806
119.644	120.000	299.110
120.000	120.000	109.000
130.000	120.000	114.000
160.000	120.000	114.000
180.000	120.000	114.000
200.000	120.000	114.000
230.000	120.000	114.000
240.000	120.000	109.000
240.356	120.000	600.890
127.635	130.000	319.087
130.000	130.000	119.000
160.000	130.000	119.000
180.000	130.000	119.000
200.000	130.000	119.000
230.000	130.000	119.000
232.365	130.000	580.913
180.000	136.347	.000

Factor of Safety = 3.263

SURFACE # 3

Coordinate X	Coordinate Y	Coordinate Z
180.000	22.640	.000
137.565	40.000	343.912
160.000	40.000	85.030
180.000	40.000	83.057
200.000	40.000	85.030
222.435	40.000	556.088
124.614	60.000	311.536
130.000	60.000	89.673
160.000	60.000	81.507
180.000	60.000	79.000
200.000	60.000	81.507
230.000	60.000	89.673
235.386	60.000	588.464
111.664	80.000	279.159
120.000	80.000	101.000
130.000	80.000	104.000
160.000	80.000	94.851
180.000	80.000	91.949
200.000	80.000	94.851
230.000	80.000	104.000
240.000	80.000	101.000
248.336	80.000	620.841
109.278	100.000	273.194
120.000	100.000	104.000
130.000	100.000	109.000
160.000	100.000	109.000
180.000	100.000	109.000
200.000	100.000	109.000
230.000	100.000	109.000
240.000	100.000	104.000

250.722	100.000	626.806
119.644	120.000	299.110
120.000	120.000	109.000
130.000	120.000	114.000
160.000	120.000	114.000
180.000	120.000	114.000
200.000	120.000	114.000
230.000	120.000	114.000
240.000	120.000	109.000
240.356	120.000	600.890
127.635	130.000	319.087
130.000	130.000	119.000
160.000	130.000	119.000
180.000	130.000	119.000
200.000	130.000	119.000
230.000	130.000	119.000
232.365	130.000	580.913
180.000	136.347	.000

Factor of Safety = 3.263

SURFACE # 4

Coordinate X	Coordinate Y	Coordinate Z
180.000	22.640	.000
137.565	40.000	343.912
160.000	40.000	85.030
180.000	40.000	83.057
200.000	40.000	85.030
222.435	40.000	556.088
124.614	60.000	311.536
130.000	60.000	89.673
160.000	60.000	81.507
180.000	60.000	79.000
200.000	60.000	81.507
230.000	60.000	89.673
235.386	60.000	588.464
111.664	80.000	279.159
120.000	80.000	101.000
130.000	80.000	104.000
160.000	80.000	94.851
180.000	80.000	91.949
200.000	80.000	94.851
230.000	80.000	104.000
240.000	80.000	101.000
248.336	80.000	620.841
109.278	100.000	273.194
120.000	100.000	104.000
130.000	100.000	109.000
160.000	100.000	109.000
180.000	100.000	109.000
200.000	100.000	109.000
230.000	100.000	109.000
240.000	100.000	104.000
250.722	100.000	626.806
119.644	120.000	299.110

120.000	120.000	109.000
130.000	120.000	114.000
160.000	120.000	114.000
180.000	120.000	114.000
200.000	120.000	114.000
230.000	120.000	114.000
240.000	120.000	109.000
240.356	120.000	600.890
127.635	130.000	319.087
130.000	130.000	119.000
160.000	130.000	119.000
180.000	130.000	119.000
200.000	130.000	119.000
230.000	130.000	119.000
232.365	130.000	580.913
180.000	136.347	.000

Factor of Safety = 3.263

SURFACE # 5

Coordinate X	Coordinate Y	Coordinate Z
180.000	22.640	.000
137.565	40.000	343.912
160.000	40.000	85.030
180.000	40.000	83.057
200.000	40.000	85.030
222.435	40.000	556.088
124.614	60.000	311.536
130.000	60.000	89.673
160.000	60.000	81.507
180.000	60.000	79.000
200.000	60.000	81.507
230.000	60.000	89.673
235.386	60.000	588.464
111.664	80.000	279.159
120.000	80.000	101.000
130.000	80.000	104.000
160.000	80.000	94.851
180.000	80.000	91.949
200.000	80.000	94.851
230.000	80.000	104.000
240.000	80.000	101.000
248.336	80.000	620.841
109.278	100.000	273.194
120.000	100.000	104.000
130.000	100.000	109.000
160.000	100.000	109.000
180.000	100.000	109.000
200.000	100.000	109.000
230.000	100.000	109.000
240.000	100.000	104.000
250.722	100.000	626.806
119.644	120.000	299.110
120.000	120.000	109.000
130.000	120.000	114.000

160.000	120.000	114.000
180.000	120.000	114.000
200.000	120.000	114.000
230.000	120.000	114.000
240.000	120.000	109.000
240.356	120.000	600.890
127.635	130.000	319.087
130.000	130.000	119.000
160.000	130.000	119.000
180.000	130.000	119.000
200.000	130.000	119.000
230.000	130.000	119.000
232.365	130.000	580.913
180.000	136.347	.000

Factor of Safety = 3.263

SURFACE # 6

Coordinate X	Coordinate Y	Coordinate Z
180.000	22.640	.000
137.565	40.000	343.912
160.000	40.000	85.030
180.000	40.000	83.057
200.000	40.000	85.030
222.435	40.000	556.088
124.614	60.000	311.536
130.000	60.000	89.673
160.000	60.000	81.507
180.000	60.000	79.000
200.000	60.000	81.507
230.000	60.000	89.673
235.386	60.000	588.464
111.664	80.000	279.159
120.000	80.000	101.000
130.000	80.000	104.000
160.000	80.000	94.851
180.000	80.000	91.949
200.000	80.000	94.851
230.000	80.000	104.000
240.000	80.000	101.000
248.336	80.000	620.841
109.278	100.000	273.194
120.000	100.000	104.000
130.000	100.000	109.000
160.000	100.000	109.000
180.000	100.000	109.000
200.000	100.000	109.000
230.000	100.000	109.000
240.000	100.000	104.000
250.722	100.000	626.806
119.644	120.000	299.110
120.000	120.000	109.000
130.000	120.000	114.000
160.000	120.000	114.000
180.000	120.000	114.000

200.000	120.000	114.000
230.000	120.000	114.000
240.000	120.000	109.000
240.356	120.000	600.890
127.635	130.000	319.087
130.000	130.000	119.000
160.000	130.000	119.000
180.000	130.000	119.000
200.000	130.000	119.000
230.000	130.000	119.000
232.365	130.000	580.913
180.000	136.347	.000

Factor of Safety = 3.263

SURFACE # 7

Coordinate X	Coordinate Y	Coordinate Z
180.000	22.640	.000
137.565	40.000	343.912
160.000	40.000	85.030
180.000	40.000	83.057
200.000	40.000	85.030
222.435	40.000	556.088
124.614	60.000	311.536
130.000	60.000	89.673
160.000	60.000	81.507
180.000	60.000	79.000
200.000	60.000	81.507
230.000	60.000	89.673
235.386	60.000	588.464
111.664	80.000	279.159
120.000	80.000	101.000
130.000	80.000	104.000
160.000	80.000	94.851
180.000	80.000	91.949
200.000	80.000	94.851
230.000	80.000	104.000
240.000	80.000	101.000
248.336	80.000	620.841
109.278	100.000	273.194
120.000	100.000	104.000
130.000	100.000	109.000
160.000	100.000	109.000
180.000	100.000	109.000
200.000	100.000	109.000
230.000	100.000	109.000
240.000	100.000	104.000
250.722	100.000	626.806
119.644	120.000	299.110
120.000	120.000	109.000
130.000	120.000	114.000
160.000	120.000	114.000
180.000	120.000	114.000
200.000	120.000	114.000
230.000	120.000	114.000

240.000	120.000	109.000
240.356	120.000	600.890
127.635	130.000	319.087
130.000	130.000	119.000
160.000	130.000	119.000
180.000	130.000	119.000
200.000	130.000	119.000
230.000	130.000	119.000
232.365	130.000	580.913
180.000	136.347	.000

Factor of Safety = 3.263

SURFACE # 8

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	294.422
120.000	100.000	101.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	101.000
242.231	100.000	605.578
118.093	120.000	295.233
120.000	120.000	104.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	104.000
241.907	120.000	604.767
119.407	130.000	298.518
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
240.593	130.000	601.482

129.127	140.000	322.818
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	577.182
140.000	160.000	350.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	550.000
180.000	160.653	.000

Factor of Safety = 4.117

SURFACE # 9

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	294.422
120.000	100.000	101.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	101.000
242.231	100.000	605.578
118.093	120.000	295.233
120.000	120.000	104.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	104.000
241.907	120.000	604.767
119.407	130.000	298.518
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000

230.000	130.000	114.000
240.000	130.000	109.000
240.593	130.000	601.482
129.127	140.000	322.818
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	577.182
140.000	160.000	350.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	550.000
180.000	160.653	.000

Factor of Safety = 4.117

SURFACE # 10

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	294.422
120.000	100.000	101.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	101.000
242.231	100.000	605.578
118.093	120.000	295.233
120.000	120.000	104.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	104.000
241.907	120.000	604.767
119.407	130.000	298.518
120.000	130.000	109.000
130.000	130.000	114.000

160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
240.593	130.000	601.482
129.127	140.000	322.818
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	577.182
140.000	160.000	350.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	550.000
180.000	160.653	.000

Factor of Safety = 4.117

Example 2

Geometry of the Example 2 is shown on Figure B2. The following soil parameters were used:

Unit weight of soil   100 pcf
Cohesion   500 psf
Friction angle   0 deg

Figure B2 Illustration for Example 2

PCSTBL3D Version 2.0

[illegible]

```

*****
*           PC-STABL3D           *
*  3-DIMENSIONAL SLOPE STABILITY PROGRAM  *
*           FROM PURDUE UNIVERSITY           *
*           VERSION 2.1           *
*****

```

Number of X coordinates : 15
 Number of Y coordinates : 13
 Number of soils : 1

X COORDINATES :

coordinate X(1) = 40.00
 coordinate X(2) = 60.00
 coordinate X(3) = 80.00
 coordinate X(4) = 100.00
 coordinate X(5) = 120.00
 coordinate X(6) = 130.00
 coordinate X(7) = 160.00
 coordinate X(8) = 180.00
 coordinate X(9) = 200.00
 coordinate X(10) = 230.00
 coordinate X(11) = 240.00
 coordinate X(12) = 260.00
 coordinate X(13) = 280.00
 coordinate X(14) = 300.00
 coordinate X(15) = 320.00

Y COORDINATES :

coordinate Y(1) = .00
 coordinate Y(2) = 20.00
 coordinate Y(3) = 40.00
 coordinate Y(4) = 60.00
 coordinate Y(5) = 80.00
 coordinate Y(6) = 100.00
 coordinate Y(7) = 120.00
 coordinate Y(8) = 130.00
 coordinate Y(9) = 140.00
 coordinate Y(10) = 160.00
 coordinate Y(11) = 180.00
 coordinate Y(12) = 200.00
 coordinate Y(13) = 230.00

COORDINATES Z FOR SOIL # 1

Z (1, 1) = 100.00   Z (1, 2) = 100.00   Z (1, 3) = 100.00
 Z (1, 4) = 106.67   Z (1, 5) = 113.33   Z (1, 6) = 120.00
 Z (1, 7) = 126.67   Z (1, 8) = 130.00   Z (1, 9) = 130.00
 Z (1,10) = 130.00   Z (1,11) = 130.00   Z (1,12) = 130.00
 Z (1,13) = 130.00   Z (2, 1) = 100.00   Z (2, 2) = 100.00
 Z (2, 3) = 100.00   Z (2, 4) = 106.67   Z (2, 5) = 113.33
 Z (2, 6) = 120.00   Z (2, 7) = 126.67   Z (2, 8) = 130.00
 Z (2, 9) = 130.00   Z (2,10) = 130.00   Z (2,11) = 130.00
 Z (2,12) = 130.00   Z (2,13) = 130.00   Z (3, 1) = 100.00
 Z (3, 2) = 100.00   Z (3, 3) = 100.00   Z (3, 4) = 106.67

$Z(3, 5) = 113.33$ $Z(3, 6) = 120.00$ $Z(3, 7) = 126.67$
 $Z(3, 8) = 130.00$ $Z(3, 9) = 130.00$ $Z(3, 10) = 130.00$
 $Z(3, 11) = 130.00$ $Z(3, 12) = 130.00$ $Z(3, 13) = 130.00$
 $Z(4, 1) = 100.00$ $Z(4, 2) = 100.00$ $Z(4, 3) = 100.00$
 $Z(4, 4) = 106.67$ $Z(4, 5) = 113.33$ $Z(4, 6) = 120.00$
 $Z(4, 7) = 126.67$ $Z(4, 8) = 130.00$ $Z(4, 9) = 130.00$
 $Z(4, 10) = 130.00$ $Z(4, 11) = 130.00$ $Z(4, 12) = 130.00$
 $Z(4, 13) = 130.00$ $Z(5, 1) = 100.00$ $Z(5, 2) = 100.00$
 $Z(5, 3) = 100.00$ $Z(5, 4) = 106.67$ $Z(5, 5) = 113.33$
 $Z(5, 6) = 120.00$ $Z(5, 7) = 126.67$ $Z(5, 8) = 130.00$
 $Z(5, 9) = 130.00$ $Z(5, 10) = 130.00$ $Z(5, 11) = 130.00$
 $Z(5, 12) = 130.00$ $Z(5, 13) = 130.00$ $Z(6, 1) = 100.00$
 $Z(6, 2) = 100.00$ $Z(6, 3) = 100.00$ $Z(6, 4) = 106.67$
 $Z(6, 5) = 113.33$ $Z(6, 6) = 120.00$ $Z(6, 7) = 126.67$
 $Z(6, 8) = 130.00$ $Z(6, 9) = 130.00$ $Z(6, 10) = 130.00$
 $Z(6, 11) = 130.00$ $Z(6, 12) = 130.00$ $Z(6, 13) = 130.00$
 $Z(7, 1) = 100.00$ $Z(7, 2) = 100.00$ $Z(7, 3) = 100.00$
 $Z(7, 4) = 106.67$ $Z(7, 5) = 113.33$ $Z(7, 6) = 120.00$
 $Z(7, 7) = 126.67$ $Z(7, 8) = 130.00$ $Z(7, 9) = 130.00$
 $Z(7, 10) = 130.00$ $Z(7, 11) = 130.00$ $Z(7, 12) = 130.00$
 $Z(7, 13) = 130.00$ $Z(8, 1) = 100.00$ $Z(8, 2) = 100.00$
 $Z(8, 3) = 100.00$ $Z(8, 4) = 106.67$ $Z(8, 5) = 113.33$
 $Z(8, 6) = 120.00$ $Z(8, 7) = 126.67$ $Z(8, 8) = 130.00$
 $Z(8, 9) = 130.00$ $Z(8, 10) = 130.00$ $Z(8, 11) = 130.00$
 $Z(8, 12) = 130.00$ $Z(8, 13) = 130.00$ $Z(9, 1) = 100.00$
 $Z(9, 2) = 100.00$ $Z(9, 3) = 100.00$ $Z(9, 4) = 106.67$
 $Z(9, 5) = 113.33$ $Z(9, 6) = 120.00$ $Z(9, 7) = 126.67$
 $Z(9, 8) = 130.00$ $Z(9, 9) = 130.00$ $Z(9, 10) = 130.00$
 $Z(9, 11) = 130.00$ $Z(9, 12) = 130.00$ $Z(9, 13) = 130.00$
 $Z(10, 1) = 100.00$ $Z(10, 2) = 100.00$ $Z(10, 3) = 100.00$
 $Z(10, 4) = 106.67$ $Z(10, 5) = 113.33$ $Z(10, 6) = 120.00$
 $Z(10, 7) = 126.67$ $Z(10, 8) = 130.00$ $Z(10, 9) = 130.00$
 $Z(10, 10) = 130.00$ $Z(10, 11) = 130.00$ $Z(10, 12) = 130.00$
 $Z(10, 13) = 130.00$ $Z(11, 1) = 100.00$ $Z(11, 2) = 100.00$
 $Z(11, 3) = 100.00$ $Z(11, 4) = 106.67$ $Z(11, 5) = 113.33$
 $Z(11, 6) = 120.00$ $Z(11, 7) = 126.67$ $Z(11, 8) = 130.00$
 $Z(11, 9) = 130.00$ $Z(11, 10) = 130.00$ $Z(11, 11) = 130.00$
 $Z(11, 12) = 130.00$ $Z(11, 13) = 130.00$ $Z(12, 1) = 100.00$
 $Z(12, 2) = 100.00$ $Z(12, 3) = 100.00$ $Z(12, 4) = 106.67$
 $Z(12, 5) = 113.33$ $Z(12, 6) = 120.00$ $Z(12, 7) = 126.67$
 $Z(12, 8) = 130.00$ $Z(12, 9) = 130.00$ $Z(12, 10) = 130.00$
 $Z(12, 11) = 130.00$ $Z(12, 12) = 130.00$ $Z(12, 13) = 130.00$
 $Z(13, 1) = 100.00$ $Z(13, 2) = 100.00$ $Z(13, 3) = 100.00$
 $Z(13, 4) = 106.67$ $Z(13, 5) = 113.33$ $Z(13, 6) = 120.00$
 $Z(13, 7) = 126.67$ $Z(13, 8) = 130.00$ $Z(13, 9) = 130.00$
 $Z(13, 10) = 130.00$ $Z(13, 11) = 130.00$ $Z(13, 12) = 130.00$
 $Z(13, 13) = 130.00$ $Z(14, 1) = 100.00$ $Z(14, 2) = 100.00$
 $Z(14, 3) = 100.00$ $Z(14, 4) = 106.67$ $Z(14, 5) = 113.33$
 $Z(14, 6) = 120.00$ $Z(14, 7) = 126.67$ $Z(14, 8) = 130.00$
 $Z(14, 9) = 130.00$ $Z(14, 10) = 130.00$ $Z(14, 11) = 130.00$
 $Z(14, 12) = 130.00$ $Z(14, 13) = 130.00$ $Z(15, 1) = 100.00$
 $Z(15, 2) = 100.00$ $Z(15, 3) = 100.00$ $Z(15, 4) = 106.67$
 $Z(15, 5) = 113.33$ $Z(15, 6) = 120.00$ $Z(15, 7) = 126.67$
 $Z(15, 8) = 130.00$ $Z(15, 9) = 130.00$ $Z(15, 10) = 130.00$
 $Z(15, 11) = 130.00$ $Z(15, 12) = 130.00$ $Z(15, 13) = 130.00$

BOUNDARIES FOR GENERATION IN THE Y DIRECTION :

Main Axis of Sliding was established at X = 180.00

Minimum point for initiation of polygons at Y = 20.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at Y = 110.00
 Interval for termination of polygons is = 60.00

BOUNDARIES FOR GENERATION IN THE X DIRECTION :

Coordinate Y = .00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 20.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 40.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 60.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 80.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 100.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 120.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 130.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 140.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 160.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 180.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 200.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 230.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

BOUNDARIES FOR GENERATION IN THE Z DIRECTION :

Zmin(1, 1) = 79.00	Delta Z(1, 1) = 20.00
Zmin(1, 2) = 79.00	Delta Z(1, 2) = 20.00
Zmin(1, 3) = 79.00	Delta Z(1, 3) = 20.00
Zmin(1, 4) = 79.00	Delta Z(1, 4) = 25.00
Zmin(1, 5) = 79.00	Delta Z(1, 5) = 30.00
Zmin(1, 6) = 79.00	Delta Z(1, 6) = 35.00
Zmin(1, 7) = 79.00	Delta Z(1, 7) = 40.00
Zmin(1, 8) = 79.00	Delta Z(1, 8) = 45.00
Zmin(1, 9) = 79.00	Delta Z(1, 9) = 45.00
Zmin(1,10) = 79.00	Delta Z(1,10) = 45.00
Zmin(1,11) = 79.00	Delta Z(1,11) = 45.00
Zmin(1,12) = 79.00	Delta Z(1,12) = 45.00
Zmin(1,13) = 79.00	Delta Z(1,13) = 45.00

Zmin(2, 1) =	79.00	Delta Z(2, 1) =	20.00
Zmin(2, 2) =	79.00	Delta Z(2, 2) =	20.00
Zmin(2, 3) =	79.00	Delta Z(2, 3) =	20.00
Zmin(2, 4) =	79.00	Delta Z(2, 4) =	25.00
Zmin(2, 5) =	79.00	Delta Z(2, 5) =	30.00
Zmin(2, 6) =	79.00	Delta Z(2, 6) =	35.00
Zmin(2, 7) =	79.00	Delta Z(2, 7) =	40.00
Zmin(2, 8) =	79.00	Delta Z(2, 8) =	45.00
Zmin(2, 9) =	79.00	Delta Z(2, 9) =	45.00
Zmin(2,10) =	79.00	Delta Z(2,10) =	45.00
Zmin(2,11) =	79.00	Delta Z(2,11) =	45.00
Zmin(2,12) =	79.00	Delta Z(2,12) =	45.00
Zmin(2,13) =	79.00	Delta Z(2,13) =	45.00
Zmin(3, 1) =	79.00	Delta Z(3, 1) =	20.00
Zmin(3, 2) =	79.00	Delta Z(3, 2) =	20.00
Zmin(3, 3) =	79.00	Delta Z(3, 3) =	20.00
Zmin(3, 4) =	79.00	Delta Z(3, 4) =	25.00
Zmin(3, 5) =	79.00	Delta Z(3, 5) =	30.00
Zmin(3, 6) =	79.00	Delta Z(3, 6) =	35.00
Zmin(3, 7) =	79.00	Delta Z(3, 7) =	40.00
Zmin(3, 8) =	79.00	Delta Z(3, 8) =	45.00
Zmin(3, 9) =	79.00	Delta Z(3, 9) =	45.00
Zmin(3,10) =	79.00	Delta Z(3,10) =	45.00
Zmin(3,11) =	79.00	Delta Z(3,11) =	45.00
Zmin(3,12) =	79.00	Delta Z(3,12) =	45.00
Zmin(3,13) =	79.00	Delta Z(3,13) =	45.00
Zmin(4, 1) =	79.00	Delta Z(4, 1) =	20.00
Zmin(4, 2) =	79.00	Delta Z(4, 2) =	20.00
Zmin(4, 3) =	79.00	Delta Z(4, 3) =	20.00
Zmin(4, 4) =	79.00	Delta Z(4, 4) =	25.00
Zmin(4, 5) =	79.00	Delta Z(4, 5) =	30.00
Zmin(4, 6) =	79.00	Delta Z(4, 6) =	35.00
Zmin(4, 7) =	79.00	Delta Z(4, 7) =	40.00
Zmin(4, 8) =	79.00	Delta Z(4, 8) =	45.00
Zmin(4, 9) =	79.00	Delta Z(4, 9) =	45.00
Zmin(4,10) =	79.00	Delta Z(4,10) =	45.00
Zmin(4,11) =	79.00	Delta Z(4,11) =	45.00
Zmin(4,12) =	79.00	Delta Z(4,12) =	45.00
Zmin(4,13) =	79.00	Delta Z(4,13) =	45.00
Zmin(5, 1) =	79.00	Delta Z(5, 1) =	20.00
Zmin(5, 2) =	79.00	Delta Z(5, 2) =	20.00
Zmin(5, 3) =	79.00	Delta Z(5, 3) =	20.00
Zmin(5, 4) =	79.00	Delta Z(5, 4) =	25.00
Zmin(5, 5) =	79.00	Delta Z(5, 5) =	30.00
Zmin(5, 6) =	79.00	Delta Z(5, 6) =	35.00
Zmin(5, 7) =	79.00	Delta Z(5, 7) =	40.00
Zmin(5, 8) =	79.00	Delta Z(5, 8) =	45.00
Zmin(5, 9) =	79.00	Delta Z(5, 9) =	45.00
Zmin(5,10) =	79.00	Delta Z(5,10) =	45.00
Zmin(5,11) =	79.00	Delta Z(5,11) =	45.00
Zmin(5,12) =	79.00	Delta Z(5,12) =	45.00
Zmin(5,13) =	79.00	Delta Z(5,13) =	45.00
Zmin(6, 1) =	79.00	Delta Z(6, 1) =	20.00
Zmin(6, 2) =	79.00	Delta Z(6, 2) =	20.00
Zmin(6, 3) =	79.00	Delta Z(6, 3) =	20.00
Zmin(6, 4) =	79.00	Delta Z(6, 4) =	25.00
Zmin(6, 5) =	79.00	Delta Z(6, 5) =	30.00
Zmin(6, 6) =	79.00	Delta Z(6, 6) =	35.00

Zmin(6, 7) =	79.00	Delta Z(6, 7) =	40.00
Zmin(6, 8) =	79.00	Delta Z(6, 8) =	45.00
Zmin(6, 9) =	79.00	Delta Z(6, 9) =	45.00
Zmin(6,10) =	79.00	Delta Z(6,10) =	45.00
Zmin(6,11) =	79.00	Delta Z(6,11) =	45.00
Zmin(6,12) =	79.00	Delta Z(6,12) =	45.00
Zmin(6,13) =	79.00	Delta Z(6,13) =	45.00
Zmin(7, 1) =	79.00	Delta Z(7, 1) =	20.00
Zmin(7, 2) =	79.00	Delta Z(7, 2) =	20.00
Zmin(7, 3) =	79.00	Delta Z(7, 3) =	20.00
Zmin(7, 4) =	79.00	Delta Z(7, 4) =	25.00
Zmin(7, 5) =	79.00	Delta Z(7, 5) =	30.00
Zmin(7, 6) =	79.00	Delta Z(7, 6) =	35.00
Zmin(7, 7) =	79.00	Delta Z(7, 7) =	40.00
Zmin(7, 8) =	79.00	Delta Z(7, 8) =	45.00
Zmin(7, 9) =	79.00	Delta Z(7, 9) =	45.00
Zmin(7,10) =	79.00	Delta Z(7,10) =	45.00
Zmin(7,11) =	79.00	Delta Z(7,11) =	45.00
Zmin(7,12) =	79.00	Delta Z(7,12) =	45.00
Zmin(7,13) =	79.00	Delta Z(7,13) =	45.00
Zmin(8, 1) =	79.00	Delta Z(8, 1) =	20.00
Zmin(8, 2) =	79.00	Delta Z(8, 2) =	20.00
Zmin(8, 3) =	79.00	Delta Z(8, 3) =	20.00
Zmin(8, 4) =	79.00	Delta Z(8, 4) =	25.00
Zmin(8, 5) =	79.00	Delta Z(8, 5) =	30.00
Zmin(8, 6) =	79.00	Delta Z(8, 6) =	35.00
Zmin(8, 7) =	79.00	Delta Z(8, 7) =	40.00
Zmin(8, 8) =	79.00	Delta Z(8, 8) =	45.00
Zmin(8, 9) =	79.00	Delta Z(8, 9) =	45.00
Zmin(8,10) =	79.00	Delta Z(8,10) =	45.00
Zmin(8,11) =	79.00	Delta Z(8,11) =	45.00
Zmin(8,12) =	79.00	Delta Z(8,12) =	45.00
Zmin(8,13) =	79.00	Delta Z(8,13) =	45.00
Zmin(9, 1) =	79.00	Delta Z(9, 1) =	20.00
Zmin(9, 2) =	79.00	Delta Z(9, 2) =	20.00
Zmin(9, 3) =	79.00	Delta Z(9, 3) =	20.00
Zmin(9, 4) =	79.00	Delta Z(9, 4) =	25.00
Zmin(9, 5) =	79.00	Delta Z(9, 5) =	30.00
Zmin(9, 6) =	79.00	Delta Z(9, 6) =	35.00
Zmin(9, 7) =	79.00	Delta Z(9, 7) =	40.00
Zmin(9, 8) =	79.00	Delta Z(9, 8) =	45.00
Zmin(9, 9) =	79.00	Delta Z(9, 9) =	45.00
Zmin(9,10) =	79.00	Delta Z(9,10) =	45.00
Zmin(9,11) =	79.00	Delta Z(9,11) =	45.00
Zmin(9,12) =	79.00	Delta Z(9,12) =	45.00
Zmin(9,13) =	79.00	Delta Z(9,13) =	45.00
Zmin(10, 1) =	79.00	Delta Z(10, 1) =	20.00
Zmin(10, 2) =	79.00	Delta Z(10, 2) =	20.00
Zmin(10, 3) =	79.00	Delta Z(10, 3) =	20.00
Zmin(10, 4) =	79.00	Delta Z(10, 4) =	25.00
Zmin(10, 5) =	79.00	Delta Z(10, 5) =	30.00
Zmin(10, 6) =	79.00	Delta Z(10, 6) =	35.00
Zmin(10, 7) =	79.00	Delta Z(10, 7) =	40.00
Zmin(10, 8) =	79.00	Delta Z(10, 8) =	45.00
Zmin(10, 9) =	79.00	Delta Z(10, 9) =	45.00
Zmin(10,10) =	79.00	Delta Z(10,10) =	45.00
Zmin(10,11) =	79.00	Delta Z(10,11) =	45.00
Zmin(10,12) =	79.00	Delta Z(10,12) =	45.00

Zmin(10,13) =	79.00	Delta Z(10,13) =	45.00
Zmin(11, 1) =	79.00	Delta Z(11, 1) =	20.00
Zmin(11, 2) =	79.00	Delta Z(11, 2) =	20.00
Zmin(11, 3) =	79.00	Delta Z(11, 3) =	20.00
Zmin(11, 4) =	79.00	Delta Z(11, 4) =	25.00
Zmin(11, 5) =	79.00	Delta Z(11, 5) =	30.00
Zmin(11, 6) =	79.00	Delta Z(11, 6) =	35.00
Zmin(11, 7) =	79.00	Delta Z(11, 7) =	40.00
Zmin(11, 8) =	79.00	Delta Z(11, 8) =	45.00
Zmin(11, 9) =	79.00	Delta Z(11, 9) =	45.00
Zmin(11,10) =	79.00	Delta Z(11,10) =	45.00
Zmin(11,11) =	79.00	Delta Z(11,11) =	45.00
Zmin(11,12) =	79.00	Delta Z(11,12) =	45.00
Zmin(11,13) =	79.00	Delta Z(11,13) =	45.00
Zmin(12, 1) =	79.00	Delta Z(12, 1) =	20.00
Zmin(12, 2) =	79.00	Delta Z(12, 2) =	20.00
Zmin(12, 3) =	79.00	Delta Z(12, 3) =	20.00
Zmin(12, 4) =	79.00	Delta Z(12, 4) =	25.00
Zmin(12, 5) =	79.00	Delta Z(12, 5) =	30.00
Zmin(12, 6) =	79.00	Delta Z(12, 6) =	35.00
Zmin(12, 7) =	79.00	Delta Z(12, 7) =	40.00
Zmin(12, 8) =	79.00	Delta Z(12, 8) =	45.00
Zmin(12, 9) =	79.00	Delta Z(12, 9) =	45.00
Zmin(12,10) =	79.00	Delta Z(12,10) =	45.00
Zmin(12,11) =	79.00	Delta Z(12,11) =	45.00
Zmin(12,12) =	79.00	Delta Z(12,12) =	45.00
Zmin(12,13) =	79.00	Delta Z(12,13) =	45.00
Zmin(13, 1) =	79.00	Delta Z(13, 1) =	20.00
Zmin(13, 2) =	79.00	Delta Z(13, 2) =	20.00
Zmin(13, 3) =	79.00	Delta Z(13, 3) =	20.00
Zmin(13, 4) =	79.00	Delta Z(13, 4) =	25.00
Zmin(13, 5) =	79.00	Delta Z(13, 5) =	30.00
Zmin(13, 6) =	79.00	Delta Z(13, 6) =	35.00
Zmin(13, 7) =	79.00	Delta Z(13, 7) =	40.00
Zmin(13, 8) =	79.00	Delta Z(13, 8) =	45.00
Zmin(13, 9) =	79.00	Delta Z(13, 9) =	45.00
Zmin(13,10) =	79.00	Delta Z(13,10) =	45.00
Zmin(13,11) =	79.00	Delta Z(13,11) =	45.00
Zmin(13,12) =	79.00	Delta Z(13,12) =	45.00
Zmin(13,13) =	79.00	Delta Z(13,13) =	45.00
Zmin(14, 1) =	79.00	Delta Z(14, 1) =	20.00
Zmin(14, 2) =	79.00	Delta Z(14, 2) =	20.00
Zmin(14, 3) =	79.00	Delta Z(14, 3) =	20.00
Zmin(14, 4) =	79.00	Delta Z(14, 4) =	25.00
Zmin(14, 5) =	79.00	Delta Z(14, 5) =	30.00
Zmin(14, 6) =	79.00	Delta Z(14, 6) =	35.00
Zmin(14, 7) =	79.00	Delta Z(14, 7) =	40.00
Zmin(14, 8) =	79.00	Delta Z(14, 8) =	45.00
Zmin(14, 9) =	79.00	Delta Z(14, 9) =	45.00
Zmin(14,10) =	79.00	Delta Z(14,10) =	45.00
Zmin(14,11) =	79.00	Delta Z(14,11) =	45.00
Zmin(14,12) =	79.00	Delta Z(14,12) =	45.00
Zmin(14,13) =	79.00	Delta Z(14,13) =	45.00
Zmin(15, 1) =	79.00	Delta Z(15, 1) =	20.00
Zmin(15, 2) =	79.00	Delta Z(15, 2) =	20.00
Zmin(15, 3) =	79.00	Delta Z(15, 3) =	20.00
Zmin(15, 4) =	79.00	Delta Z(15, 4) =	25.00
Zmin(15, 5) =	79.00	Delta Z(15, 5) =	30.00

Zmin(15, 6) = 79.00   Delta Z(15, 6) = 35.00
 Zmin(15, 7) = 79.00   Delta Z(15, 7) = 40.00
 Zmin(15, 8) = 79.00   Delta Z(15, 8) = 45.00
 Zmin(15, 9) = 79.00   Delta Z(15, 9) = 45.00
 Zmin(15,10) = 79.00   Delta Z(15,10) = 45.00
 Zmin(15,11) = 79.00   Delta Z(15,11) = 45.00
 Zmin(15,12) = 79.00   Delta Z(15,12) = 45.00
 Zmin(15,13) = 79.00   Delta Z(15,13) = 45.00

SOIL PROPERTIES :

Soil	Unit Weight	Cohesion	Shear Angle

1	100.00	500.0	.0

Ru of soil # 1 = .00

TEN MOST CRITICAL SURFACES :

SURFACE # 1

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.307	.000
148.097	60.000	370.244
160.000	60.000	84.276
180.000	60.000	82.207
200.000	60.000	84.276
211.903	60.000	529.756
121.131	80.000	302.828
130.000	80.000	84.292
160.000	80.000	80.698
180.000	80.000	79.154
200.000	80.000	80.698
230.000	80.000	84.292
238.869	80.000	597.172
113.743	100.000	303.324
120.000	100.000	104.000
130.000	100.000	104.000
160.000	100.000	93.758
180.000	100.000	87.428
200.000	100.000	93.758
230.000	100.000	104.000
240.000	100.000	104.000
246.257	100.000	656.706
117.008	120.000	331.512
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	109.000
180.000	120.000	103.140
200.000	120.000	109.000
230.000	120.000	109.000
240.000	120.000	109.000
242.992	120.000	688.458

119.850	130.000	359.551
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.150	130.000	720.449
132.629	140.000	420.004
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
227.371	140.000	720.026
180.000	152.146	.000

Factor of Safety = 4.096

SURFACE # 2

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.307	.000
148.097	60.000	370.244
160.000	60.000	84.276
180.000	60.000	82.207
200.000	60.000	84.276
211.903	60.000	529.756
121.131	80.000	302.828
130.000	80.000	84.292
160.000	80.000	80.698
180.000	80.000	79.154
200.000	80.000	80.698
230.000	80.000	84.292
238.869	80.000	597.172
113.743	100.000	303.324
120.000	100.000	104.000
130.000	100.000	104.000
160.000	100.000	93.758
180.000	100.000	87.428
200.000	100.000	93.758
230.000	100.000	104.000
240.000	100.000	104.000
246.257	100.000	656.706
117.008	120.000	331.512
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	109.000
180.000	120.000	103.140
200.000	120.000	109.000
230.000	120.000	109.000
240.000	120.000	109.000
242.992	120.000	688.458
119.850	130.000	359.551
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000

180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.150	130.000	720.449
132.629	140.000	420.004
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
227.371	140.000	720.026
180.000	152.146	.000

Factor of Safety = 4.096

SURFACE # 3

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000
242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000
241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000

240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 4

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000
242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000
241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598

200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 5

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000
242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000
241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000

130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 6

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000
242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000

241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 7

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000
242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381

200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000
241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 8

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000
242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000

130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000
241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 9

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	314.059
120.000	100.000	104.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	104.000

242.231	100.000	645.971
118.093	120.000	334.587
120.000	120.000	109.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	109.000
241.907	120.000	685.383
119.407	130.000	358.221
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
240.593	130.000	721.779
129.127	140.000	408.913
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	731.117
140.000	160.000	455.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	715.000
180.000	160.653	.000

Factor of Safety = 4.308

SURFACE # 10

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966

200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 4.552

Example 3

Geometry of the Example 3 is shown on Figure B3. The following soil parameters were used:

Unit weight of soil	110 pcf
Cohesion	100 psf
Friction angle	28 deg

Figure B3 Illustration for Example 3

[illegible]

```

*****
*           PC-STABL3D           *
* 3-DIMENSIONAL SLOPE STABILITY PROGRAM *
* FROM PURDUE UNIVERSITY      *
*           VERSION 2.1        *
*****

```

Number of X coordinates : 15
 Number of Y coordinates : 13
 Number of soils : 1

X COORDINATES :

coordinate X(1) = 40.00
 coordinate X(2) = 60.00
 coordinate X(3) = 80.00
 coordinate X(4) = 100.00
 coordinate X(5) = 120.00
 coordinate X(6) = 130.00
 coordinate X(7) = 160.00
 coordinate X(8) = 180.00
 coordinate X(9) = 200.00
 coordinate X(10) = 230.00
 coordinate X(11) = 240.00
 coordinate X(12) = 260.00
 coordinate X(13) = 280.00
 coordinate X(14) = 300.00
 coordinate X(15) = 320.00

Y COORDINATES :

coordinate Y(1) = .00
 coordinate Y(2) = 20.00
 coordinate Y(3) = 40.00
 coordinate Y(4) = 60.00
 coordinate Y(5) = 80.00
 coordinate Y(6) = 100.00
 coordinate Y(7) = 120.00
 coordinate Y(8) = 130.00
 coordinate Y(9) = 140.00
 coordinate Y(10) = 160.00
 coordinate Y(11) = 180.00
 coordinate Y(12) = 200.00
 coordinate Y(13) = 230.00

COORDINATES Z FOR SOIL # 1

Z (1, 1) = 100.00   Z (1, 2) = 100.00   Z (1, 3) = 100.00
 Z (1, 4) = 100.00   Z (1, 5) = 100.00   Z (1, 6) = 100.00
 Z (1, 7) = 100.00   Z (1, 8) = 100.00   Z (1, 9) = 100.00
 Z (1,10) = 100.00   Z (1,11) = 100.00   Z (1,12) = 100.00
 Z (1,13) = 100.00   Z (2, 1) = 100.00   Z (2, 2) = 100.00
 Z (2, 3) = 100.00   Z (2, 4) = 100.00   Z (2, 5) = 100.00
 Z (2, 6) = 100.00   Z (2, 7) = 103.33   Z (2, 8) = 106.67
 Z (2, 9) = 106.67   Z (2,10) = 106.67   Z (2,11) = 106.67
 Z (2,12) = 106.67   Z (2,13) = 106.67   Z (3, 1) = 100.00
 Z (3, 2) = 100.00   Z (3, 3) = 100.00   Z (3, 4) = 100.00

Z (3, 5) = 100.00   Z (3, 6) = 103.33   Z (3, 7) = 110.00
 Z (3, 8) = 113.33   Z (3, 9) = 113.33   Z (3,10) = 113.33
 Z (3,11) = 113.33   Z (3,12) = 113.33   Z (3,13) = 113.33
 Z (4, 1) = 100.00   Z (4, 2) = 100.00   Z (4, 3) = 100.00
 Z (4, 4) = 100.00   Z (4, 5) = 103.33   Z (4, 6) = 110.00
 Z (4, 7) = 116.67   Z (4, 8) = 120.00   Z (4, 9) = 120.00
 Z (4,10) = 120.00   Z (4,11) = 120.00   Z (4,12) = 120.00
 Z (4,13) = 120.00   Z (5, 1) = 100.00   Z (5, 2) = 100.00
 Z (5, 3) = 100.00   Z (5, 4) = 103.33   Z (5, 5) = 110.00
 Z (5, 6) = 116.67   Z (5, 7) = 123.33   Z (5, 8) = 126.67
 Z (5, 9) = 126.67   Z (5,10) = 126.67   Z (5,11) = 126.67
 Z (5,12) = 126.67   Z (5,13) = 126.67   Z (6, 1) = 100.00
 Z (6, 2) = 100.00   Z (6, 3) = 100.00   Z (6, 4) = 106.67
 Z (6, 5) = 113.33   Z (6, 6) = 120.00   Z (6, 7) = 126.67
 Z (6, 8) = 130.00   Z (6, 9) = 130.00   Z (6,10) = 130.00
 Z (6,11) = 130.00   Z (6,12) = 130.00   Z (6,13) = 130.00
 Z (7, 1) = 100.00   Z (7, 2) = 100.00   Z (7, 3) = 100.00
 Z (7, 4) = 106.67   Z (7, 5) = 113.33   Z (7, 6) = 120.00
 Z (7, 7) = 126.67   Z (7, 8) = 130.00   Z (7, 9) = 130.00
 Z (7,10) = 130.00   Z (7,11) = 130.00   Z (7,12) = 130.00
 Z (7,13) = 130.00   Z (8, 1) = 100.00   Z (8, 2) = 100.00
 Z (8, 3) = 100.00   Z (8, 4) = 106.67   Z (8, 5) = 113.33
 Z (8, 6) = 120.00   Z (8, 7) = 126.67   Z (8, 8) = 130.00
 Z (8, 9) = 130.00   Z (8,10) = 130.00   Z (8,11) = 130.00
 Z (8,12) = 130.00   Z (8,13) = 130.00   Z (9, 1) = 100.00
 Z (9, 2) = 100.00   Z (9, 3) = 100.00   Z (9, 4) = 106.67
 Z (9, 5) = 113.33   Z (9, 6) = 120.00   Z (9, 7) = 126.67
 Z (9, 8) = 130.00   Z (9, 9) = 130.00   Z (9,10) = 130.00
 Z (9,11) = 130.00   Z (9,12) = 130.00   Z (9,13) = 130.00
 Z (10, 1) = 100.00   Z (10, 2) = 100.00   Z (10, 3) = 100.00
 Z (10, 4) = 106.67   Z (10, 5) = 113.33   Z (10, 6) = 120.00
 Z (10, 7) = 126.67   Z (10, 8) = 130.00   Z (10, 9) = 130.00
 Z (10,10) = 130.00   Z (10,11) = 130.00   Z (10,12) = 130.00
 Z (10,13) = 130.00   Z (11, 1) = 100.00   Z (11, 2) = 100.00
 Z (11, 3) = 100.00   Z (11, 4) = 103.33   Z (11, 5) = 110.00
 Z (11, 6) = 116.67   Z (11, 7) = 123.33   Z (11, 8) = 126.67
 Z (11, 9) = 126.67   Z (11,10) = 126.67   Z (11,11) = 126.67
 Z (11,12) = 126.67   Z (11,13) = 126.67   Z (12, 1) = 100.00
 Z (12, 2) = 100.00   Z (12, 3) = 100.00   Z (12, 4) = 100.00
 Z (12, 5) = 103.33   Z (12, 6) = 110.00   Z (12, 7) = 116.67
 Z (12, 8) = 120.00   Z (12, 9) = 120.00   Z (12,10) = 120.00
 Z (12,11) = 120.00   Z (12,12) = 120.00   Z (12,13) = 120.00
 Z (13, 1) = 100.00   Z (13, 2) = 100.00   Z (13, 3) = 100.00
 Z (13, 4) = 100.00   Z (13, 5) = 100.00   Z (13, 6) = 103.33
 Z (13, 7) = 110.00   Z (13, 8) = 113.33   Z (13, 9) = 113.33
 Z (13,10) = 113.33   Z (13,11) = 113.33   Z (13,12) = 113.33
 Z (13,13) = 113.33   Z (14, 1) = 100.00   Z (14, 2) = 100.00
 Z (14, 3) = 100.00   Z (14, 4) = 100.00   Z (14, 5) = 100.00
 Z (14, 6) = 100.00   Z (14, 7) = 103.33   Z (14, 8) = 106.67
 Z (14, 9) = 106.67   Z (14,10) = 106.67   Z (14,11) = 106.67
 Z (14,12) = 106.67   Z (14,13) = 106.67   Z (15, 1) = 100.00
 Z (15, 2) = 100.00   Z (15, 3) = 100.00   Z (15, 4) = 100.00
 Z (15, 5) = 100.00   Z (15, 6) = 100.00   Z (15, 7) = 100.00
 Z (15, 8) = 100.00   Z (15, 9) = 100.00   Z (15,10) = 100.00
 Z (15,11) = 100.00   Z (15,12) = 100.00   Z (15,13) = 100.00

BOUNDARIES FOR GENERATION IN THE Y DIRECTION :

Main Axis of Sliding was established at $X = 180.00$

Minimum point for initiation of polygons at $Y = 20.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $Y = 110.00$
 Interval for termination of polygons is $= 60.00$

BOUNDARIES FOR GENERATION IN THE X DIRECTION :

Coordinate $Y = .00$
 Minimum point for initiation of polygons at $X = 120.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 200.00$
 Interval for termination of polygons is $= 40.00$

Coordinate $Y = 20.00$
 Minimum point for initiation of polygons at $X = 120.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 200.00$
 Interval for termination of polygons is $= 40.00$

Coordinate $Y = 40.00$
 Minimum point for initiation of polygons at $X = 120.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 200.00$
 Interval for termination of polygons is $= 40.00$

Coordinate $Y = 60.00$
 Minimum point for initiation of polygons at $X = 100.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 220.00$
 Interval for termination of polygons is $= 40.00$

Coordinate $Y = 80.00$
 Minimum point for initiation of polygons at $X = 100.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 220.00$
 Interval for termination of polygons is $= 40.00$

Coordinate $Y = 100.00$
 Minimum point for initiation of polygons at $X = 100.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 220.00$
 Interval for termination of polygons is $= 40.00$

Coordinate $Y = 120.00$
 Minimum point for initiation of polygons at $X = 100.00$
 Interval for initiation of polygons is $= 40.00$
 Minimum point for termination of polygons at $X = 220.00$
 Interval for termination of polygons is $= 40.00$

Coordinate Y = 130.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 140.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 160.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 180.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 200.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 230.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

BOUNDARIES FOR GENERATION IN THE Z DIRECTION :

Zmin(1, 1) = 79.00	Delta Z(1, 1) = 20.00
Zmin(1, 2) = 79.00	Delta Z(1, 2) = 20.00
Zmin(1, 3) = 79.00	Delta Z(1, 3) = 20.00
Zmin(1, 4) = 79.00	Delta Z(1, 4) = 20.00
Zmin(1, 5) = 79.00	Delta Z(1, 5) = 20.00
Zmin(1, 6) = 79.00	Delta Z(1, 6) = 20.00
Zmin(1, 7) = 79.00	Delta Z(1, 7) = 20.00
Zmin(1, 8) = 79.00	Delta Z(1, 8) = 20.00
Zmin(1, 9) = 79.00	Delta Z(1, 9) = 20.00
Zmin(1,10) = 79.00	Delta Z(1,10) = 20.00
Zmin(1,11) = 79.00	Delta Z(1,11) = 20.00
Zmin(1,12) = 79.00	Delta Z(1,12) = 20.00
Zmin(1,13) = 79.00	Delta Z(1,13) = 20.00

Zmin(2, 1) =	79.00	Delta Z(2, 1) =	20.00
Zmin(2, 2) =	79.00	Delta Z(2, 2) =	20.00
Zmin(2, 3) =	79.00	Delta Z(2, 3) =	20.00
Zmin(2, 4) =	79.00	Delta Z(2, 4) =	20.00
Zmin(2, 5) =	79.00	Delta Z(2, 5) =	20.00
Zmin(2, 6) =	79.00	Delta Z(2, 6) =	20.00
Zmin(2, 7) =	79.00	Delta Z(2, 7) =	21.00
Zmin(2, 8) =	79.00	Delta Z(2, 8) =	24.00
Zmin(2, 9) =	79.00	Delta Z(2, 9) =	24.00
Zmin(2,10) =	79.00	Delta Z(2,10) =	24.00
Zmin(2,11) =	79.00	Delta Z(2,11) =	24.00
Zmin(2,12) =	79.00	Delta Z(2,12) =	24.00
Zmin(2,13) =	79.00	Delta Z(2,13) =	24.00
Zmin(3, 1) =	79.00	Delta Z(3, 1) =	20.00
Zmin(3, 2) =	79.00	Delta Z(3, 2) =	20.00
Zmin(3, 3) =	79.00	Delta Z(3, 3) =	20.00
Zmin(3, 4) =	79.00	Delta Z(3, 4) =	20.00
Zmin(3, 5) =	79.00	Delta Z(3, 5) =	20.00
Zmin(3, 6) =	79.00	Delta Z(3, 6) =	22.00
Zmin(3, 7) =	79.00	Delta Z(3, 7) =	25.00
Zmin(3, 8) =	79.00	Delta Z(3, 8) =	30.00
Zmin(3, 9) =	79.00	Delta Z(3, 9) =	30.00
Zmin(3,10) =	79.00	Delta Z(3,10) =	30.00
Zmin(3,11) =	79.00	Delta Z(3,11) =	30.00
Zmin(3,12) =	79.00	Delta Z(3,12) =	30.00
Zmin(3,13) =	79.00	Delta Z(3,13) =	30.00
Zmin(4, 1) =	79.00	Delta Z(4, 1) =	20.00
Zmin(4, 2) =	79.00	Delta Z(4, 2) =	20.00
Zmin(4, 3) =	79.00	Delta Z(4, 3) =	20.00
Zmin(4, 4) =	79.00	Delta Z(4, 4) =	20.00
Zmin(4, 5) =	79.00	Delta Z(4, 5) =	22.00
Zmin(4, 6) =	79.00	Delta Z(4, 6) =	25.00
Zmin(4, 7) =	79.00	Delta Z(4, 7) =	32.00
Zmin(4, 8) =	79.00	Delta Z(4, 8) =	35.00
Zmin(4, 9) =	79.00	Delta Z(4, 9) =	35.00
Zmin(4,10) =	79.00	Delta Z(4,10) =	35.00
Zmin(4,11) =	79.00	Delta Z(4,11) =	35.00
Zmin(4,12) =	79.00	Delta Z(4,12) =	35.00
Zmin(4,13) =	79.00	Delta Z(4,13) =	35.00
Zmin(5, 1) =	79.00	Delta Z(5, 1) =	20.00
Zmin(5, 2) =	79.00	Delta Z(5, 2) =	20.00
Zmin(5, 3) =	79.00	Delta Z(5, 3) =	20.00
Zmin(5, 4) =	79.00	Delta Z(5, 4) =	22.00
Zmin(5, 5) =	79.00	Delta Z(5, 5) =	25.00
Zmin(5, 6) =	79.00	Delta Z(5, 6) =	30.00
Zmin(5, 7) =	79.00	Delta Z(5, 7) =	35.00
Zmin(5, 8) =	79.00	Delta Z(5, 8) =	35.00
Zmin(5, 9) =	79.00	Delta Z(5, 9) =	35.00
Zmin(5,10) =	79.00	Delta Z(5,10) =	35.00
Zmin(5,11) =	79.00	Delta Z(5,11) =	35.00
Zmin(5,12) =	79.00	Delta Z(5,12) =	35.00
Zmin(5,13) =	79.00	Delta Z(5,13) =	35.00
Zmin(6, 1) =	79.00	Delta Z(6, 1) =	20.00
Zmin(6, 2) =	79.00	Delta Z(6, 2) =	20.00
Zmin(6, 3) =	79.00	Delta Z(6, 3) =	20.00
Zmin(6, 4) =	79.00	Delta Z(6, 4) =	25.00
Zmin(6, 5) =	79.00	Delta Z(6, 5) =	30.00
Zmin(6, 6) =	79.00	Delta Z(6, 6) =	35.00

Zmin(6, 7) =	79.00	Delta Z(6, 7) =	40.00
Zmin(6, 8) =	79.00	Delta Z(6, 8) =	45.00
Zmin(6, 9) =	79.00	Delta Z(6, 9) =	45.00
Zmin(6,10) =	79.00	Delta Z(6,10) =	45.00
Zmin(6,11) =	79.00	Delta Z(6,11) =	45.00
Zmin(6,12) =	79.00	Delta Z(6,12) =	45.00
Zmin(6,13) =	79.00	Delta Z(6,13) =	45.00
Zmin(7, 1) =	79.00	Delta Z(7, 1) =	20.00
Zmin(7, 2) =	79.00	Delta Z(7, 2) =	20.00
Zmin(7, 3) =	79.00	Delta Z(7, 3) =	20.00
Zmin(7, 4) =	79.00	Delta Z(7, 4) =	25.00
Zmin(7, 5) =	79.00	Delta Z(7, 5) =	30.00
Zmin(7, 6) =	79.00	Delta Z(7, 6) =	35.00
Zmin(7, 7) =	79.00	Delta Z(7, 7) =	40.00
Zmin(7, 8) =	79.00	Delta Z(7, 8) =	45.00
Zmin(7, 9) =	79.00	Delta Z(7, 9) =	45.00
Zmin(7,10) =	79.00	Delta Z(7,10) =	45.00
Zmin(7,11) =	79.00	Delta Z(7,11) =	45.00
Zmin(7,12) =	79.00	Delta Z(7,12) =	45.00
Zmin(7,13) =	79.00	Delta Z(7,13) =	45.00
Zmin(8, 1) =	79.00	Delta Z(8, 1) =	20.00
Zmin(8, 2) =	79.00	Delta Z(8, 2) =	20.00
Zmin(8, 3) =	79.00	Delta Z(8, 3) =	20.00
Zmin(8, 4) =	79.00	Delta Z(8, 4) =	25.00
Zmin(8, 5) =	79.00	Delta Z(8, 5) =	30.00
Zmin(8, 6) =	79.00	Delta Z(8, 6) =	35.00
Zmin(8, 7) =	79.00	Delta Z(8, 7) =	40.00
Zmin(8, 8) =	79.00	Delta Z(8, 8) =	45.00
Zmin(8, 9) =	79.00	Delta Z(8, 9) =	45.00
Zmin(8,10) =	79.00	Delta Z(8,10) =	45.00
Zmin(8,11) =	79.00	Delta Z(8,11) =	45.00
Zmin(8,12) =	79.00	Delta Z(8,12) =	45.00
Zmin(8,13) =	79.00	Delta Z(8,13) =	45.00
Zmin(9, 1) =	79.00	Delta Z(9, 1) =	20.00
Zmin(9, 2) =	79.00	Delta Z(9, 2) =	20.00
Zmin(9, 3) =	79.00	Delta Z(9, 3) =	20.00
Zmin(9, 4) =	79.00	Delta Z(9, 4) =	25.00
Zmin(9, 5) =	79.00	Delta Z(9, 5) =	30.00
Zmin(9, 6) =	79.00	Delta Z(9, 6) =	35.00
Zmin(9, 7) =	79.00	Delta Z(9, 7) =	40.00
Zmin(9, 8) =	79.00	Delta Z(9, 8) =	45.00
Zmin(9, 9) =	79.00	Delta Z(9, 9) =	45.00
Zmin(9,10) =	79.00	Delta Z(9,10) =	45.00
Zmin(9,11) =	79.00	Delta Z(9,11) =	45.00
Zmin(9,12) =	79.00	Delta Z(9,12) =	45.00
Zmin(9,13) =	79.00	Delta Z(9,13) =	45.00
Zmin(10, 1) =	79.00	Delta Z(10, 1) =	20.00
Zmin(10, 2) =	79.00	Delta Z(10, 2) =	20.00
Zmin(10, 3) =	79.00	Delta Z(10, 3) =	20.00
Zmin(10, 4) =	79.00	Delta Z(10, 4) =	25.00
Zmin(10, 5) =	79.00	Delta Z(10, 5) =	30.00
Zmin(10, 6) =	79.00	Delta Z(10, 6) =	35.00
Zmin(10, 7) =	79.00	Delta Z(10, 7) =	40.00
Zmin(10, 8) =	79.00	Delta Z(10, 8) =	45.00
Zmin(10, 9) =	79.00	Delta Z(10, 9) =	45.00
Zmin(10,10) =	79.00	Delta Z(10,10) =	45.00
Zmin(10,11) =	79.00	Delta Z(10,11) =	45.00
Zmin(10,12) =	79.00	Delta Z(10,12) =	45.00

Zmin(10,13) =	79.00	Delta Z(10,13) =	45.00
Zmin(11, 1) =	79.00	Delta Z(11, 1) =	20.00
Zmin(11, 2) =	79.00	Delta Z(11, 2) =	20.00
Zmin(11, 3) =	79.00	Delta Z(11, 3) =	20.00
Zmin(11, 4) =	79.00	Delta Z(11, 4) =	22.00
Zmin(11, 5) =	79.00	Delta Z(11, 5) =	25.00
Zmin(11, 6) =	79.00	Delta Z(11, 6) =	30.00
Zmin(11, 7) =	79.00	Delta Z(11, 7) =	35.00
Zmin(11, 8) =	79.00	Delta Z(11, 8) =	35.00
Zmin(11, 9) =	79.00	Delta Z(11, 9) =	35.00
Zmin(11,10) =	79.00	Delta Z(11,10) =	35.00
Zmin(11,11) =	79.00	Delta Z(11,11) =	35.00
Zmin(11,12) =	79.00	Delta Z(11,12) =	35.00
Zmin(11,13) =	79.00	Delta Z(11,13) =	35.00
Zmin(12, 1) =	79.00	Delta Z(12, 1) =	20.00
Zmin(12, 2) =	79.00	Delta Z(12, 2) =	20.00
Zmin(12, 3) =	79.00	Delta Z(12, 3) =	20.00
Zmin(12, 4) =	79.00	Delta Z(12, 4) =	20.00
Zmin(12, 5) =	79.00	Delta Z(12, 5) =	22.00
Zmin(12, 6) =	79.00	Delta Z(12, 6) =	25.00
Zmin(12, 7) =	79.00	Delta Z(12, 7) =	32.00
Zmin(12, 8) =	79.00	Delta Z(12, 8) =	35.00
Zmin(12, 9) =	79.00	Delta Z(12, 9) =	35.00
Zmin(12,10) =	79.00	Delta Z(12,10) =	35.00
Zmin(12,11) =	79.00	Delta Z(12,11) =	35.00
Zmin(12,12) =	79.00	Delta Z(12,12) =	35.00
Zmin(12,13) =	79.00	Delta Z(12,13) =	35.00
Zmin(13, 1) =	79.00	Delta Z(13, 1) =	20.00
Zmin(13, 2) =	79.00	Delta Z(13, 2) =	20.00
Zmin(13, 3) =	79.00	Delta Z(13, 3) =	20.00
Zmin(13, 4) =	79.00	Delta Z(13, 4) =	20.00
Zmin(13, 5) =	79.00	Delta Z(13, 5) =	20.00
Zmin(13, 6) =	79.00	Delta Z(13, 6) =	22.00
Zmin(13, 7) =	79.00	Delta Z(13, 7) =	25.00
Zmin(13, 8) =	79.00	Delta Z(13, 8) =	30.00
Zmin(13, 9) =	79.00	Delta Z(13, 9) =	30.00
Zmin(13,10) =	79.00	Delta Z(13,10) =	30.00
Zmin(13,11) =	79.00	Delta Z(13,11) =	30.00
Zmin(13,12) =	79.00	Delta Z(13,12) =	30.00
Zmin(13,13) =	79.00	Delta Z(13,13) =	30.00
Zmin(14, 1) =	79.00	Delta Z(14, 1) =	20.00
Zmin(14, 2) =	79.00	Delta Z(14, 2) =	20.00
Zmin(14, 3) =	79.00	Delta Z(14, 3) =	20.00
Zmin(14, 4) =	79.00	Delta Z(14, 4) =	20.00
Zmin(14, 5) =	79.00	Delta Z(14, 5) =	20.00
Zmin(14, 6) =	79.00	Delta Z(14, 6) =	20.00
Zmin(14, 7) =	79.00	Delta Z(14, 7) =	21.00
Zmin(14, 8) =	79.00	Delta Z(14, 8) =	24.00
Zmin(14, 9) =	79.00	Delta Z(14, 9) =	24.00
Zmin(14,10) =	79.00	Delta Z(14,10) =	24.00
Zmin(14,11) =	79.00	Delta Z(14,11) =	24.00
Zmin(14,12) =	79.00	Delta Z(14,12) =	24.00
Zmin(14,13) =	79.00	Delta Z(14,13) =	24.00
Zmin(15, 1) =	79.00	Delta Z(15, 1) =	20.00
Zmin(15, 2) =	79.00	Delta Z(15, 2) =	20.00
Zmin(15, 3) =	79.00	Delta Z(15, 3) =	20.00
Zmin(15, 4) =	79.00	Delta Z(15, 4) =	20.00
Zmin(15, 5) =	79.00	Delta Z(15, 5) =	20.00

Zmin(15, 6) = 79.00   Delta Z(15, 6) = 20.00
 Zmin(15, 7) = 79.00   Delta Z(15, 7) = 20.00
 Zmin(15, 8) = 79.00   Delta Z(15, 8) = 20.00
 Zmin(15, 9) = 79.00   Delta Z(15, 9) = 20.00
 Zmin(15,10) = 79.00   Delta Z(15,10) = 20.00
 Zmin(15,11) = 79.00   Delta Z(15,11) = 20.00
 Zmin(15,12) = 79.00   Delta Z(15,12) = 20.00
 Zmin(15,13) = 79.00   Delta Z(15,13) = 20.00

SOIL PROPERTIES :

Soil	Unit Weight	Cohesion	Shear Angle
1	110.00	100.0	28.0

Ru of soil # 1 = .00

TEN MOST CRITICAL SURFACES :

SURFACE # 1

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.307	.000
148.097	60.000	370.244
160.000	60.000	84.276
180.000	60.000	82.207
200.000	60.000	84.276
211.903	60.000	529.756
121.131	80.000	302.828
130.000	80.000	84.292
160.000	80.000	80.698
180.000	80.000	79.154
200.000	80.000	80.698
230.000	80.000	84.292
238.869	80.000	597.172
113.743	100.000	284.358
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	93.758
180.000	100.000	87.428
200.000	100.000	93.758
230.000	100.000	104.000
240.000	100.000	101.000
246.257	100.000	615.642
117.008	120.000	292.519
120.000	120.000	104.000
130.000	120.000	109.000
160.000	120.000	109.000
180.000	120.000	103.140
200.000	120.000	109.000
230.000	120.000	109.000
240.000	120.000	104.000
242.992	120.000	607.481

119.850	130.000	299.626
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
240.150	130.000	600.374
132.629	140.000	331.574
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
227.371	140.000	568.426
180.000	152.146	.000

Factor of Safety = 1.603

SURFACE # 2

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.307	.000
148.097	60.000	370.244
160.000	60.000	84.276
180.000	60.000	82.207
200.000	60.000	84.276
211.903	60.000	529.756
121.131	80.000	302.828
130.000	80.000	84.292
160.000	80.000	80.698
180.000	80.000	79.154
200.000	80.000	80.698
230.000	80.000	84.292
238.869	80.000	597.172
113.743	100.000	284.358
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	93.758
180.000	100.000	87.428
200.000	100.000	93.758
230.000	100.000	104.000
240.000	100.000	101.000
246.257	100.000	615.642
117.008	120.000	292.519
120.000	120.000	104.000
130.000	120.000	109.000
160.000	120.000	109.000
180.000	120.000	103.140
200.000	120.000	109.000
230.000	120.000	109.000
240.000	120.000	104.000
242.992	120.000	607.481
119.850	130.000	299.626
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000

180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
240.150	130.000	600.374
132.629	140.000	331.574
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
227.371	140.000	568.426
180.000	152.146	.000

Factor of Safety = 1.603

SURFACE # 3

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.307	.000
148.097	60.000	370.244
160.000	60.000	84.276
180.000	60.000	82.207
200.000	60.000	84.276
211.903	60.000	529.756
121.131	80.000	302.828
130.000	80.000	84.292
160.000	80.000	80.698
180.000	80.000	79.154
200.000	80.000	80.698
230.000	80.000	84.292
238.869	80.000	597.172
113.743	100.000	284.358
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	93.758
180.000	100.000	87.428
200.000	100.000	93.758
230.000	100.000	104.000
240.000	100.000	101.000
246.257	100.000	615.642
117.008	120.000	292.519
120.000	120.000	104.000
130.000	120.000	109.000
160.000	120.000	109.000
180.000	120.000	103.140
200.000	120.000	109.000
230.000	120.000	109.000
240.000	120.000	104.000
242.992	120.000	607.481
119.850	130.000	299.626
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000

240.150	130.000	600.374
132.629	140.000	331.574
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
227.371	140.000	568.426
180.000	152.146	.000

Factor of Safety = 1.603

SURFACE # 4

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.307	.000
148.097	60.000	370.244
160.000	60.000	84.276
180.000	60.000	82.207
200.000	60.000	84.276
211.903	60.000	529.756
121.131	80.000	302.828
130.000	80.000	84.292
160.000	80.000	80.698
180.000	80.000	79.154
200.000	80.000	80.698
230.000	80.000	84.292
238.869	80.000	597.172
113.743	100.000	284.358
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	93.758
180.000	100.000	87.428
200.000	100.000	93.758
230.000	100.000	104.000
240.000	100.000	101.000
246.257	100.000	615.642
117.008	120.000	292.519
120.000	120.000	104.000
130.000	120.000	109.000
160.000	120.000	109.000
180.000	120.000	103.140
200.000	120.000	109.000
230.000	120.000	109.000
240.000	120.000	104.000
242.992	120.000	607.481
119.850	130.000	299.626
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
240.150	130.000	600.374
132.629	140.000	331.574
160.000	140.000	119.000
180.000	140.000	119.000

200.000	140.000	119.000
227.371	140.000	568.426
180.000	152.146	.000

Factor of Safety = 1.603

SURFACE # 5

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.370	.000
138.225	60.000	345.562
160.000	60.000	95.803
180.000	60.000	95.549
200.000	60.000	95.803
221.775	60.000	554.438
125.056	80.000	312.640
130.000	80.000	84.907
160.000	80.000	82.613
180.000	80.000	82.346
200.000	80.000	82.613
230.000	80.000	84.907
234.944	80.000	587.360
114.484	100.000	286.209
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	92.738
180.000	100.000	84.590
200.000	100.000	92.738
230.000	100.000	104.000
240.000	100.000	101.000
245.516	100.000	613.791
116.914	120.000	292.284
120.000	120.000	104.000
130.000	120.000	108.457
160.000	120.000	104.945
180.000	120.000	103.570
200.000	120.000	104.945
230.000	120.000	108.457
240.000	120.000	104.000
243.086	120.000	607.716
118.573	130.000	296.433
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
241.427	130.000	603.567
180.000	137.337	.000

Factor of Safety = 1.819

SURFACE # 6

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.370	.000
138.225	60.000	345.562
160.000	60.000	95.803
180.000	60.000	95.549
200.000	60.000	95.803
221.775	60.000	554.438
125.056	80.000	312.640
130.000	80.000	84.907
160.000	80.000	82.613
180.000	80.000	82.346
200.000	80.000	82.613
230.000	80.000	84.907
234.944	80.000	587.360
114.484	100.000	286.209
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	92.738
180.000	100.000	84.590
200.000	100.000	92.738
230.000	100.000	104.000
240.000	100.000	101.000
245.516	100.000	613.791
116.914	120.000	292.284
120.000	120.000	104.000
130.000	120.000	108.457
160.000	120.000	104.945
180.000	120.000	103.570
200.000	120.000	104.945
230.000	120.000	108.457
240.000	120.000	104.000
243.086	120.000	607.716
118.573	130.000	296.433
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
241.427	130.000	603.567
180.000	137.337	.000

Factor of Safety = 1.819

SURFACE # 7

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.370	.000
138.225	60.000	345.562
160.000	60.000	95.803
180.000	60.000	95.549
200.000	60.000	95.803
221.775	60.000	554.438
125.056	80.000	312.640

130.000	80.000	84.907
160.000	80.000	82.613
180.000	80.000	82.346
200.000	80.000	82.613
230.000	80.000	84.907
234.944	80.000	587.360
114.484	100.000	286.209
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	92.738
180.000	100.000	84.590
200.000	100.000	92.738
230.000	100.000	104.000
240.000	100.000	101.000
245.516	100.000	613.791
116.914	120.000	292.284
120.000	120.000	104.000
130.000	120.000	108.457
160.000	120.000	104.945
180.000	120.000	103.570
200.000	120.000	104.945
230.000	120.000	108.457
240.000	120.000	104.000
243.086	120.000	607.716
118.573	130.000	296.433
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
241.427	130.000	603.567
180.000	137.337	.000

Factor of Safety = 1.819

SURFACE # 8

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.370	.000
138.225	60.000	345.562
160.000	60.000	95.803
180.000	60.000	95.549
200.000	60.000	95.803
221.775	60.000	554.438
125.056	80.000	312.640
130.000	80.000	84.907
160.000	80.000	82.613
180.000	80.000	82.346
200.000	80.000	82.613
230.000	80.000	84.907
234.944	80.000	587.360
114.484	100.000	286.209
120.000	100.000	101.000
130.000	100.000	104.000

160.000	100.000	92.738
180.000	100.000	84.590
200.000	100.000	92.738
230.000	100.000	104.000
240.000	100.000	101.000
245.516	100.000	613.791
116.914	120.000	292.284
120.000	120.000	104.000
130.000	120.000	108.457
160.000	120.000	104.945
180.000	120.000	103.570
200.000	120.000	104.945
230.000	120.000	108.457
240.000	120.000	104.000
243.086	120.000	607.716
118.573	130.000	296.433
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
241.427	130.000	603.567
180.000	137.337	.000

Factor of Safety = 1.819

SURFACE # 9

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.370	.000
138.225	60.000	345.562
160.000	60.000	95.803
180.000	60.000	95.549
200.000	60.000	95.803
221.775	60.000	554.438
125.056	80.000	312.640
130.000	80.000	84.907
160.000	80.000	82.613
180.000	80.000	82.346
200.000	80.000	82.613
230.000	80.000	84.907
234.944	80.000	587.360
114.484	100.000	286.209
120.000	100.000	101.000
130.000	100.000	104.000
160.000	100.000	92.738
180.000	100.000	84.590
200.000	100.000	92.738
230.000	100.000	104.000
240.000	100.000	101.000
245.516	100.000	613.791
116.914	120.000	292.284
120.000	120.000	104.000
130.000	120.000	108.457

160.000	120.000	104.945
180.000	120.000	103.570
200.000	120.000	104.945
230.000	120.000	108.457
240.000	120.000	104.000
243.086	120.000	607.716
118.573	130.000	296.433
120.000	130.000	109.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
241.427	130.000	603.567
180.000	137.337	.000

Factor of Safety = 1.819

SURFACE # 10

Coordinate X	Coordinate Y	Coordinate Z
180.000	46.742	.000
153.769	60.000	384.422
160.000	60.000	94.802
180.000	60.000	91.850
200.000	60.000	94.802
206.231	60.000	515.578
125.274	80.000	313.185
130.000	80.000	90.641
160.000	80.000	83.934
180.000	80.000	83.876
200.000	80.000	83.934
230.000	80.000	90.641
234.726	80.000	586.815
117.769	100.000	294.422
120.000	100.000	101.000
130.000	100.000	101.471
160.000	100.000	88.691
180.000	100.000	87.211
200.000	100.000	88.691
230.000	100.000	101.471
240.000	100.000	101.000
242.231	100.000	605.578
118.093	120.000	295.233
120.000	120.000	104.000
130.000	120.000	107.775
160.000	120.000	103.387
180.000	120.000	103.381
200.000	120.000	103.387
230.000	120.000	107.775
240.000	120.000	104.000
241.907	120.000	604.767
119.407	130.000	298.518
120.000	130.000	109.000
130.000	130.000	114.000

160.000	130.000	114.000
180.000	130.000	111.598
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	109.000
240.593	130.000	601.482
129.127	140.000	322.818
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
230.873	140.000	577.182
140.000	160.000	350.000
160.000	160.000	124.000
180.000	160.000	124.000
200.000	160.000	124.000
220.000	160.000	550.000
180.000	160.653	.000

Factor of Safety = 1.923

Example 4

Geometry of the Example 4 is shown on Figure B4. The following soil parameters were used:

Unit weight of soil   110 pcf
Cohesion   100 psf
Friction angle   28 deg .

Figure B4   Illustration for Example 4

PCSTBL3D Version 2.0

[illegible]

```

*****
*           PC-STABL3D           *
*  3-DIMENSIONAL SLOPE STABILITY PROGRAM  *
*           FROM PURDUE UNIVERSITY           *
*           VERSION 2.1           *
*****

```

Number of X coordinates : 15
 Number of Y coordinates : 13
 Number of soils : 1

X COORDINATES :

coordinate X(1) = 40.00
 coordinate X(2) = 60.00
 coordinate X(3) = 80.00
 coordinate X(4) = 100.00
 coordinate X(5) = 120.00
 coordinate X(6) = 130.00
 coordinate X(7) = 160.00
 coordinate X(8) = 180.00
 coordinate X(9) = 200.00
 coordinate X(10) = 230.00
 coordinate X(11) = 240.00
 coordinate X(12) = 260.00
 coordinate X(13) = 280.00
 coordinate X(14) = 300.00
 coordinate X(15) = 320.00

Y COORDINATES :

coordinate Y(1) = .00
 coordinate Y(2) = 20.00
 coordinate Y(3) = 40.00
 coordinate Y(4) = 60.00
 coordinate Y(5) = 80.00
 coordinate Y(6) = 100.00
 coordinate Y(7) = 120.00
 coordinate Y(8) = 130.00
 coordinate Y(9) = 140.00
 coordinate Y(10) = 160.00
 coordinate Y(11) = 180.00
 coordinate Y(12) = 200.00
 coordinate Y(13) = 230.00

COORDINATES Z FOR SOIL # 1

Z (1, 1) = 100.00   Z (1, 2) = 100.00   Z (1, 3) = 100.00
 Z (1, 4) = 106.67   Z (1, 5) = 113.33   Z (1, 6) = 120.00
 Z (1, 7) = 126.67   Z (1, 8) = 130.00   Z (1, 9) = 130.00
 Z (1,10) = 130.00   Z (1,11) = 130.00   Z (1,12) = 130.00
 Z (1,13) = 130.00   Z (2, 1) = 100.00   Z (2, 2) = 100.00
 Z (2, 3) = 100.00   Z (2, 4) = 106.67   Z (2, 5) = 113.33
 Z (2, 6) = 120.00   Z (2, 7) = 126.67   Z (2, 8) = 130.00
 Z (2, 9) = 130.00   Z (2,10) = 130.00   Z (2,11) = 130.00
 Z (2,12) = 130.00   Z (2,13) = 130.00   Z (3, 1) = 100.00
 Z (3, 2) = 100.00   Z (3, 3) = 100.00   Z (3, 4) = 106.67

Z (3, 5) = 113.33	Z (3, 6) = 120.00	Z (3, 7) = 126.67
Z (3, 8) = 130.00	Z (3, 9) = 130.00	Z (3,10) = 130.00
Z (3,11) = 130.00	Z (3,12) = 130.00	Z (3,13) = 130.00
Z (4, 1) = 100.00	Z (4, 2) = 100.00	Z (4, 3) = 100.00
Z (4, 4) = 106.67	Z (4, 5) = 113.33	Z (4, 6) = 120.00
Z (4, 7) = 126.67	Z (4, 8) = 130.00	Z (4, 9) = 130.00
Z (4,10) = 130.00	Z (4,11) = 130.00	Z (4,12) = 130.00
Z (4,13) = 130.00	Z (5, 1) = 100.00	Z (5, 2) = 100.00
Z (5, 3) = 100.00	Z (5, 4) = 106.67	Z (5, 5) = 113.33
Z (5, 6) = 120.00	Z (5, 7) = 126.67	Z (5, 8) = 130.00
Z (5, 9) = 130.00	Z (5,10) = 130.00	Z (5,11) = 130.00
Z (5,12) = 130.00	Z (5,13) = 130.00	Z (6, 1) = 100.00
Z (6, 2) = 100.00	Z (6, 3) = 100.00	Z (6, 4) = 106.67
Z (6, 5) = 113.33	Z (6, 6) = 120.00	Z (6, 7) = 126.67
Z (6, 8) = 130.00	Z (6, 9) = 130.00	Z (6,10) = 130.00
Z (6,11) = 130.00	Z (6,12) = 130.00	Z (6,13) = 130.00
Z (7, 1) = 100.00	Z (7, 2) = 100.00	Z (7, 3) = 100.00
Z (7, 4) = 106.67	Z (7, 5) = 113.33	Z (7, 6) = 120.00
Z (7, 7) = 126.67	Z (7, 8) = 130.00	Z (7, 9) = 130.00
Z (7,10) = 130.00	Z (7,11) = 130.00	Z (7,12) = 130.00
Z (7,13) = 130.00	Z (8, 1) = 100.00	Z (8, 2) = 100.00
Z (8, 3) = 100.00	Z (8, 4) = 106.67	Z (8, 5) = 113.33
Z (8, 6) = 120.00	Z (8, 7) = 126.67	Z (8, 8) = 130.00
Z (8, 9) = 130.00	Z (8,10) = 130.00	Z (8,11) = 130.00
Z (8,12) = 130.00	Z (8,13) = 130.00	Z (9, 1) = 100.00
Z (9, 2) = 100.00	Z (9, 3) = 100.00	Z (9, 4) = 106.67
Z (9, 5) = 113.33	Z (9, 6) = 120.00	Z (9, 7) = 126.67
Z (9, 8) = 130.00	Z (9, 9) = 130.00	Z (9,10) = 130.00
Z (9,11) = 130.00	Z (9,12) = 130.00	Z (9,13) = 130.00
Z (10, 1) = 100.00	Z (10, 2) = 100.00	Z (10, 3) = 100.00
Z (10, 4) = 106.67	Z (10, 5) = 113.33	Z (10, 6) = 120.00
Z (10, 7) = 126.67	Z (10, 8) = 130.00	Z (10, 9) = 130.00
Z (10,10) = 130.00	Z (10,11) = 130.00	Z (10,12) = 130.00
Z (10,13) = 130.00	Z (11, 1) = 100.00	Z (11, 2) = 100.00
Z (11, 3) = 100.00	Z (11, 4) = 106.67	Z (11, 5) = 113.33
Z (11, 6) = 120.00	Z (11, 7) = 126.67	Z (11, 8) = 130.00
Z (11, 9) = 130.00	Z (11,10) = 130.00	Z (11,11) = 130.00
Z (11,12) = 130.00	Z (11,13) = 130.00	Z (12, 1) = 100.00
Z (12, 2) = 100.00	Z (12, 3) = 100.00	Z (12, 4) = 106.67
Z (12, 5) = 113.33	Z (12, 6) = 120.00	Z (12, 7) = 126.67
Z (12, 8) = 130.00	Z (12, 9) = 130.00	Z (12,10) = 130.00
Z (12,11) = 130.00	Z (12,12) = 130.00	Z (12,13) = 130.00
Z (13, 1) = 100.00	Z (13, 2) = 100.00	Z (13, 3) = 100.00
Z (13, 4) = 106.67	Z (13, 5) = 113.33	Z (13, 6) = 120.00
Z (13, 7) = 126.67	Z (13, 8) = 130.00	Z (13, 9) = 130.00
Z (13,10) = 130.00	Z (13,11) = 130.00	Z (13,12) = 130.00
Z (13,13) = 130.00	Z (14, 1) = 100.00	Z (14, 2) = 100.00
Z (14, 3) = 100.00	Z (14, 4) = 106.67	Z (14, 5) = 113.33
Z (14, 6) = 120.00	Z (14, 7) = 126.67	Z (14, 8) = 130.00
Z (14, 9) = 130.00	Z (14,10) = 130.00	Z (14,11) = 130.00
Z (14,12) = 130.00	Z (14,13) = 130.00	Z (15, 1) = 100.00
Z (15, 2) = 100.00	Z (15, 3) = 100.00	Z (15, 4) = 106.67
Z (15, 5) = 113.33	Z (15, 6) = 120.00	Z (15, 7) = 126.67
Z (15, 8) = 130.00	Z (15, 9) = 130.00	Z (15,10) = 130.00
Z (15,11) = 130.00	Z (15,12) = 130.00	Z (15,13) = 130.00

BOUNDARIES FOR GENERATION IN THE Y DIRECTION :

Main Axis of Sliding was established at X = 180.00

Minimum point for initiation of polygons at Y = 20.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at Y = 110.00
 Interval for termination of polygons is = 60.00

BOUNDARIES FOR GENERATION IN THE X DIRECTION :

Coordinate Y = .00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 20.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 40.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 60.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 80.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 100.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 120.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 130.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 140.00
 Minimum point for initiation of polygons at X = 100.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 220.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 160.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 180.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 200.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

Coordinate Y = 230.00
 Minimum point for initiation of polygons at X = 120.00
 Interval for initiation of polygons is = 40.00
 Minimum point for termination of polygons at X = 200.00
 Interval for termination of polygons is = 40.00

BOUNDARIES FOR GENERATION IN THE Z DIRECTION :

Zmin(1, 1) = 79.00	Delta Z(1, 1) = 20.00
Zmin(1, 2) = 79.00	Delta Z(1, 2) = 20.00
Zmin(1, 3) = 79.00	Delta Z(1, 3) = 20.00
Zmin(1, 4) = 79.00	Delta Z(1, 4) = 25.00
Zmin(1, 5) = 79.00	Delta Z(1, 5) = 30.00
Zmin(1, 6) = 79.00	Delta Z(1, 6) = 35.00
Zmin(1, 7) = 79.00	Delta Z(1, 7) = 40.00
Zmin(1, 8) = 79.00	Delta Z(1, 8) = 45.00
Zmin(1, 9) = 79.00	Delta Z(1, 9) = 45.00
Zmin(1,10) = 79.00	Delta Z(1,10) = 45.00
Zmin(1,11) = 79.00	Delta Z(1,11) = 45.00
Zmin(1,12) = 79.00	Delta Z(1,12) = 45.00
Zmin(1,13) = 79.00	Delta Z(1,13) = 45.00

Zmin(2, 1) =	79.00	Delta Z(2, 1) =	20.00
Zmin(2, 2) =	79.00	Delta Z(2, 2) =	20.00
Zmin(2, 3) =	79.00	Delta Z(2, 3) =	20.00
Zmin(2, 4) =	79.00	Delta Z(2, 4) =	25.00
Zmin(2, 5) =	79.00	Delta Z(2, 5) =	30.00
Zmin(2, 6) =	79.00	Delta Z(2, 6) =	35.00
Zmin(2, 7) =	79.00	Delta Z(2, 7) =	40.00
Zmin(2, 8) =	79.00	Delta Z(2, 8) =	45.00
Zmin(2, 9) =	79.00	Delta Z(2, 9) =	45.00
Zmin(2,10) =	79.00	Delta Z(2,10) =	45.00
Zmin(2,11) =	79.00	Delta Z(2,11) =	45.00
Zmin(2,12) =	79.00	Delta Z(2,12) =	45.00
Zmin(2,13) =	79.00	Delta Z(2,13) =	45.00
Zmin(3, 1) =	79.00	Delta Z(3, 1) =	20.00
Zmin(3, 2) =	79.00	Delta Z(3, 2) =	20.00
Zmin(3, 3) =	79.00	Delta Z(3, 3) =	20.00
Zmin(3, 4) =	79.00	Delta Z(3, 4) =	25.00
Zmin(3, 5) =	79.00	Delta Z(3, 5) =	30.00
Zmin(3, 6) =	79.00	Delta Z(3, 6) =	35.00
Zmin(3, 7) =	79.00	Delta Z(3, 7) =	40.00
Zmin(3, 8) =	79.00	Delta Z(3, 8) =	45.00
Zmin(3, 9) =	79.00	Delta Z(3, 9) =	45.00
Zmin(3,10) =	79.00	Delta Z(3,10) =	45.00
Zmin(3,11) =	79.00	Delta Z(3,11) =	45.00
Zmin(3,12) =	79.00	Delta Z(3,12) =	45.00
Zmin(3,13) =	79.00	Delta Z(3,13) =	45.00
Zmin(4, 1) =	79.00	Delta Z(4, 1) =	20.00
Zmin(4, 2) =	79.00	Delta Z(4, 2) =	20.00
Zmin(4, 3) =	79.00	Delta Z(4, 3) =	20.00
Zmin(4, 4) =	79.00	Delta Z(4, 4) =	25.00
Zmin(4, 5) =	79.00	Delta Z(4, 5) =	30.00
Zmin(4, 6) =	79.00	Delta Z(4, 6) =	35.00
Zmin(4, 7) =	79.00	Delta Z(4, 7) =	40.00
Zmin(4, 8) =	79.00	Delta Z(4, 8) =	45.00
Zmin(4, 9) =	79.00	Delta Z(4, 9) =	45.00
Zmin(4,10) =	79.00	Delta Z(4,10) =	45.00
Zmin(4,11) =	79.00	Delta Z(4,11) =	45.00
Zmin(4,12) =	79.00	Delta Z(4,12) =	45.00
Zmin(4,13) =	79.00	Delta Z(4,13) =	45.00
Zmin(5, 1) =	79.00	Delta Z(5, 1) =	20.00
Zmin(5, 2) =	79.00	Delta Z(5, 2) =	20.00
Zmin(5, 3) =	79.00	Delta Z(5, 3) =	20.00
Zmin(5, 4) =	79.00	Delta Z(5, 4) =	25.00
Zmin(5, 5) =	79.00	Delta Z(5, 5) =	30.00
Zmin(5, 6) =	79.00	Delta Z(5, 6) =	35.00
Zmin(5, 7) =	79.00	Delta Z(5, 7) =	40.00
Zmin(5, 8) =	79.00	Delta Z(5, 8) =	45.00
Zmin(5, 9) =	79.00	Delta Z(5, 9) =	45.00
Zmin(5,10) =	79.00	Delta Z(5,10) =	45.00
Zmin(5,11) =	79.00	Delta Z(5,11) =	45.00
Zmin(5,12) =	79.00	Delta Z(5,12) =	45.00
Zmin(5,13) =	79.00	Delta Z(5,13) =	45.00
Zmin(6, 1) =	79.00	Delta Z(6, 1) =	20.00
Zmin(6, 2) =	79.00	Delta Z(6, 2) =	20.00
Zmin(6, 3) =	79.00	Delta Z(6, 3) =	20.00
Zmin(6, 4) =	79.00	Delta Z(6, 4) =	25.00
Zmin(6, 5) =	79.00	Delta Z(6, 5) =	30.00
Zmin(6, 6) =	79.00	Delta Z(6, 6) =	35.00

Zmin(6, 7) =	79.00	Delta Z(6, 7) =	40.00
Zmin(6, 8) =	79.00	Delta Z(6, 8) =	45.00
Zmin(6, 9) =	79.00	Delta Z(6, 9) =	45.00
Zmin(6,10) =	79.00	Delta Z(6,10) =	45.00
Zmin(6,11) =	79.00	Delta Z(6,11) =	45.00
Zmin(6,12) =	79.00	Delta Z(6,12) =	45.00
Zmin(6,13) =	79.00	Delta Z(6,13) =	45.00
Zmin(7, 1) =	79.00	Delta Z(7, 1) =	20.00
Zmin(7, 2) =	79.00	Delta Z(7, 2) =	20.00
Zmin(7, 3) =	79.00	Delta Z(7, 3) =	20.00
Zmin(7, 4) =	79.00	Delta Z(7, 4) =	25.00
Zmin(7, 5) =	79.00	Delta Z(7, 5) =	30.00
Zmin(7, 6) =	79.00	Delta Z(7, 6) =	35.00
Zmin(7, 7) =	79.00	Delta Z(7, 7) =	40.00
Zmin(7, 8) =	79.00	Delta Z(7, 8) =	45.00
Zmin(7, 9) =	79.00	Delta Z(7, 9) =	45.00
Zmin(7,10) =	79.00	Delta Z(7,10) =	45.00
Zmin(7,11) =	79.00	Delta Z(7,11) =	45.00
Zmin(7,12) =	79.00	Delta Z(7,12) =	45.00
Zmin(7,13) =	79.00	Delta Z(7,13) =	45.00
Zmin(8, 1) =	79.00	Delta Z(8, 1) =	20.00
Zmin(8, 2) =	79.00	Delta Z(8, 2) =	20.00
Zmin(8, 3) =	79.00	Delta Z(8, 3) =	20.00
Zmin(8, 4) =	79.00	Delta Z(8, 4) =	25.00
Zmin(8, 5) =	79.00	Delta Z(8, 5) =	30.00
Zmin(8, 6) =	79.00	Delta Z(8, 6) =	35.00
Zmin(8, 7) =	79.00	Delta Z(8, 7) =	40.00
Zmin(8, 8) =	79.00	Delta Z(8, 8) =	45.00
Zmin(8, 9) =	79.00	Delta Z(8, 9) =	45.00
Zmin(8,10) =	79.00	Delta Z(8,10) =	45.00
Zmin(8,11) =	79.00	Delta Z(8,11) =	45.00
Zmin(8,12) =	79.00	Delta Z(8,12) =	45.00
Zmin(8,13) =	79.00	Delta Z(8,13) =	45.00
Zmin(9, 1) =	79.00	Delta Z(9, 1) =	20.00
Zmin(9, 2) =	79.00	Delta Z(9, 2) =	20.00
Zmin(9, 3) =	79.00	Delta Z(9, 3) =	20.00
Zmin(9, 4) =	79.00	Delta Z(9, 4) =	25.00
Zmin(9, 5) =	79.00	Delta Z(9, 5) =	30.00
Zmin(9, 6) =	79.00	Delta Z(9, 6) =	35.00
Zmin(9, 7) =	79.00	Delta Z(9, 7) =	40.00
Zmin(9, 8) =	79.00	Delta Z(9, 8) =	45.00
Zmin(9, 9) =	79.00	Delta Z(9, 9) =	45.00
Zmin(9,10) =	79.00	Delta Z(9,10) =	45.00
Zmin(9,11) =	79.00	Delta Z(9,11) =	45.00
Zmin(9,12) =	79.00	Delta Z(9,12) =	45.00
Zmin(9,13) =	79.00	Delta Z(9,13) =	45.00
Zmin(10, 1) =	79.00	Delta Z(10, 1) =	20.00
Zmin(10, 2) =	79.00	Delta Z(10, 2) =	20.00
Zmin(10, 3) =	79.00	Delta Z(10, 3) =	20.00
Zmin(10, 4) =	79.00	Delta Z(10, 4) =	25.00
Zmin(10, 5) =	79.00	Delta Z(10, 5) =	30.00
Zmin(10, 6) =	79.00	Delta Z(10, 6) =	35.00
Zmin(10, 7) =	79.00	Delta Z(10, 7) =	40.00
Zmin(10, 8) =	79.00	Delta Z(10, 8) =	45.00
Zmin(10, 9) =	79.00	Delta Z(10, 9) =	45.00
Zmin(10,10) =	79.00	Delta Z(10,10) =	45.00
Zmin(10,11) =	79.00	Delta Z(10,11) =	45.00
Zmin(10,12) =	79.00	Delta Z(10,12) =	45.00

Zmin(10,13) =	79.00	Delta Z(10,13) =	45.00
Zmin(11, 1) =	79.00	Delta Z(11, 1) =	20.00
Zmin(11, 2) =	79.00	Delta Z(11, 2) =	20.00
Zmin(11, 3) =	79.00	Delta Z(11, 3) =	20.00
Zmin(11, 4) =	79.00	Delta Z(11, 4) =	25.00
Zmin(11, 5) =	79.00	Delta Z(11, 5) =	30.00
Zmin(11, 6) =	79.00	Delta Z(11, 6) =	35.00
Zmin(11, 7) =	79.00	Delta Z(11, 7) =	40.00
Zmin(11, 8) =	79.00	Delta Z(11, 8) =	45.00
Zmin(11, 9) =	79.00	Delta Z(11, 9) =	45.00
Zmin(11,10) =	79.00	Delta Z(11,10) =	45.00
Zmin(11,11) =	79.00	Delta Z(11,11) =	45.00
Zmin(11,12) =	79.00	Delta Z(11,12) =	45.00
Zmin(11,13) =	79.00	Delta Z(11,13) =	45.00
Zmin(12, 1) =	79.00	Delta Z(12, 1) =	20.00
Zmin(12, 2) =	79.00	Delta Z(12, 2) =	20.00
Zmin(12, 3) =	79.00	Delta Z(12, 3) =	20.00
Zmin(12, 4) =	79.00	Delta Z(12, 4) =	25.00
Zmin(12, 5) =	79.00	Delta Z(12, 5) =	30.00
Zmin(12, 6) =	79.00	Delta Z(12, 6) =	35.00
Zmin(12, 7) =	79.00	Delta Z(12, 7) =	40.00
Zmin(12, 8) =	79.00	Delta Z(12, 8) =	45.00
Zmin(12, 9) =	79.00	Delta Z(12, 9) =	45.00
Zmin(12,10) =	79.00	Delta Z(12,10) =	45.00
Zmin(12,11) =	79.00	Delta Z(12,11) =	45.00
Zmin(12,12) =	79.00	Delta Z(12,12) =	45.00
Zmin(12,13) =	79.00	Delta Z(12,13) =	45.00
Zmin(13, 1) =	79.00	Delta Z(13, 1) =	20.00
Zmin(13, 2) =	79.00	Delta Z(13, 2) =	20.00
Zmin(13, 3) =	79.00	Delta Z(13, 3) =	20.00
Zmin(13, 4) =	79.00	Delta Z(13, 4) =	25.00
Zmin(13, 5) =	79.00	Delta Z(13, 5) =	30.00
Zmin(13, 6) =	79.00	Delta Z(13, 6) =	35.00
Zmin(13, 7) =	79.00	Delta Z(13, 7) =	40.00
Zmin(13, 8) =	79.00	Delta Z(13, 8) =	45.00
Zmin(13, 9) =	79.00	Delta Z(13, 9) =	45.00
Zmin(13,10) =	79.00	Delta Z(13,10) =	45.00
Zmin(13,11) =	79.00	Delta Z(13,11) =	45.00
Zmin(13,12) =	79.00	Delta Z(13,12) =	45.00
Zmin(13,13) =	79.00	Delta Z(13,13) =	45.00
Zmin(14, 1) =	79.00	Delta Z(14, 1) =	20.00
Zmin(14, 2) =	79.00	Delta Z(14, 2) =	20.00
Zmin(14, 3) =	79.00	Delta Z(14, 3) =	20.00
Zmin(14, 4) =	79.00	Delta Z(14, 4) =	25.00
Zmin(14, 5) =	79.00	Delta Z(14, 5) =	30.00
Zmin(14, 6) =	79.00	Delta Z(14, 6) =	35.00
Zmin(14, 7) =	79.00	Delta Z(14, 7) =	40.00
Zmin(14, 8) =	79.00	Delta Z(14, 8) =	45.00
Zmin(14, 9) =	79.00	Delta Z(14, 9) =	45.00
Zmin(14,10) =	79.00	Delta Z(14,10) =	45.00
Zmin(14,11) =	79.00	Delta Z(14,11) =	45.00
Zmin(14,12) =	79.00	Delta Z(14,12) =	45.00
Zmin(14,13) =	79.00	Delta Z(14,13) =	45.00
Zmin(15, 1) =	79.00	Delta Z(15, 1) =	20.00
Zmin(15, 2) =	79.00	Delta Z(15, 2) =	20.00
Zmin(15, 3) =	79.00	Delta Z(15, 3) =	20.00
Zmin(15, 4) =	79.00	Delta Z(15, 4) =	25.00
Zmin(15, 5) =	79.00	Delta Z(15, 5) =	30.00

Zmin(15, 6) = 79.00   Delta Z(15, 6) = 35.00
 Zmin(15, 7) = 79.00   Delta Z(15, 7) = 40.00
 Zmin(15, 8) = 79.00   Delta Z(15, 8) = 45.00
 Zmin(15, 9) = 79.00   Delta Z(15, 9) = 45.00
 Zmin(15,10) = 79.00   Delta Z(15,10) = 45.00
 Zmin(15,11) = 79.00   Delta Z(15,11) = 45.00
 Zmin(15,12) = 79.00   Delta Z(15,12) = 45.00
 Zmin(15,13) = 79.00   Delta Z(15,13) = 45.00

SOIL PROPERTIES :

Soil	Unit Weight	Cohesion	Shear Angle
1	110.00	100.0	28.0

Ru of soil # 1 = .00

TEN MOST CRITICAL SURFACES :

SURFACE # 1

Coordinate X	Coordinate Y	Coordinate Z
180.000	40.370	.000
138.225	60.000	345.562
160.000	60.000	95.803
180.000	60.000	95.549
200.000	60.000	95.803
221.775	60.000	554.438
125.056	80.000	312.640
130.000	80.000	84.907
160.000	80.000	82.613
180.000	80.000	82.346
200.000	80.000	82.613
230.000	80.000	84.907
234.944	80.000	587.360
114.484	100.000	305.299
120.000	100.000	104.000
130.000	100.000	104.000
160.000	100.000	92.738
180.000	100.000	84.590
200.000	100.000	92.738
230.000	100.000	104.000
240.000	100.000	104.000
245.516	100.000	654.731
116.914	120.000	331.246
120.000	120.000	109.000
130.000	120.000	108.457
160.000	120.000	104.945
180.000	120.000	103.570
200.000	120.000	104.945
230.000	120.000	108.457
240.000	120.000	109.000
243.086	120.000	688.724

118.573	130.000	355.719
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	114.000
180.000	130.000	114.000
200.000	130.000	114.000
230.000	130.000	114.000
240.000	130.000	114.000
241.427	130.000	724.281
180.000	137.337	.000

Factor of Safety = 1.822

SURFACE # 2

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935

128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 3

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000

160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 4

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000

200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 5

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000

231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 6

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 7

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 8

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 9

Coordinate X	Coordinate Y	Coordinate Z
180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

SURFACE # 10

Coordinate X	Coordinate Y	Coordinate Z

180.000	43.397	.000
150.678	60.000	376.696
160.000	60.000	88.254
180.000	60.000	86.833
200.000	60.000	88.254
209.322	60.000	523.304
122.693	80.000	306.733
130.000	80.000	97.993
160.000	80.000	83.627
180.000	80.000	82.713
200.000	80.000	83.627
230.000	80.000	97.993
237.307	80.000	593.267
114.992	100.000	306.656
120.000	100.000	99.357
130.000	100.000	89.323
160.000	100.000	88.130
180.000	100.000	87.966
200.000	100.000	88.130
230.000	100.000	89.323
240.000	100.000	99.357
245.008	100.000	653.374
116.531	120.000	330.162
120.000	120.000	109.000
130.000	120.000	109.000
160.000	120.000	107.418
180.000	120.000	102.401
200.000	120.000	107.418
230.000	120.000	109.000
240.000	120.000	109.000
243.469	120.000	689.808
118.355	130.000	355.065
120.000	130.000	114.000
130.000	130.000	114.000
160.000	130.000	112.261
180.000	130.000	112.238
200.000	130.000	112.261
230.000	130.000	114.000
240.000	130.000	114.000
241.645	130.000	724.935
128.958	140.000	408.377
130.000	140.000	119.000
160.000	140.000	119.000
180.000	140.000	119.000
200.000	140.000	119.000
230.000	140.000	119.000
231.042	140.000	731.653
180.000	156.018	.000

Factor of Safety = 1.946

APPENDIX C   PROGRAM SOURCE CODE

PCSTABL3D

^H

PAGE 1
04-12-91
12:06:11

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1 $nottruncate
2 $title:'PCSTBL3D v2.0'
3 $large
4     PROGRAM THESIS
5 *
6 *===== Prologue =====
7 *
8 * Purpose:
9 *     Main module of the program PCSTBL3D
10 *
11 * History:
12 *   Version      Programmer      Date      Description
13 *   -----
14 *   1.0          J.E.Thomas      created
15 *   2.0          G.Moshkovich    code beautified,
16 *                                     variables declared
17 *
18 * IN args/commons      Units      Description
19 * -----
20 *
21 * OUT args/commons      Units      Description
22 * -----
23 *
24 * INOUT args/commons    Units      Description
25 * -----
26 *
27 * Processing:
28 *     Main processing module. Gets in/output datafile names, number
29 * of critical surfaces to be generated, and invoke calculation routines.
30 *
31 * Special Requirements:
32 *     none
33 *
34 *----- Include files -----
35 $include:'CURRENTF.INC'
**** Begin listing of: CURRENTF.INC
1 *
2 *===== Prologue =====
3 *
4 * Purpose:
5 *     Include file with parameter declaration for the program PCSTL3D
6 *
7 * History:
8 *   Version      Programmer      Date      Description
9 *   -----
10 *   1.0          G.Moshkovich    created
11 *
12 * IN args/commons      Units      Description
13 * -----
14 *
15 * OUT args/commons      Units      Description
16 * -----
17 *
18 * INOUT args/commons    Units      Description
19 * -----
20 *
21 * Processing:
22 *     none

```

^H PCSTBL3D v2.0

 PAGE 2
 04-12-91
 12:06:11

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

23 *
24 * Special Requirements:
25 *   none
26 *
27 *----- Include files -----
28 *----- Constant declarations -----
29     integer MCOORD, MSOIL, NUMCOL, NCORN
30
31     parameter (MCOORD = 25, MSOIL = 5)
32     parameter (NUMCOL = 10, NCORN = 4)
33     PARAMETER (MAXSUR=2000)
34
35 *----- MCOORD - maximum number of points in each direction
36 *----- MSOIL - maximum number of soil layers
37 *----- NUMCOL - maximum number of generated columns
38 *----- NCORN - number of nods for each column (do not change)
39
40 *----- Argument declarations -----
41 *----- Global/External declarations -----
42 *----- Local declarations -----
43
44
45
**** End listing of: CURRENTF.INC
36 *----- Constant declarations -----
37 *----- Argument declarations -----
38 *----- Global/External declarations -----
39     EXTERNAL INDATA,GENSUR,PARAM1,CRITIC
40 *----- Local declarations -----
41     REAL X(MCOORD),Y(MCOORD),Z(MSOIL,MCOORD,MCOORD),XMIN1(MCOORD),
42     1DXMIN1(MCOORD),XMIN2(MCOORD),DXMIN2(MCOORD),ZMIN(MCOORD,MCOORD),
43     2DZMIN(MCOORD,MCOORD),GAMA(MSOIL),COES(MSOIL),FI(MSOIL),RU(MSOIL),
44     3UZ(MSOIL,MCOORD,MCOORD),XIPOL(MCOORD),XFPOL(MCOORD),XPOL(MCOORD,
45     4MCOORD),YPOL(MCOORD),ZPOL(MCOORD,MCOORD),XDIFF(MCOORD),
46     5XCOL(MCOORD,-NUMCOL:NUMCOL,NCORN),YCOL(MCOORD,-NUMCOL:NUMCOL,
47     6NCORN),ZCOL(MCOORD,-NUMCOL:NUMCOL,NCORN),NCOLEF(MCOORD),
48     7NPTLEF(MCOORD),NCORIG(MCOORD),NPTRIG(MCOORD),ALPHAXZ(MCOORD,-
49     8NUMCOL:NUMCOL),ALPHAYZ(MCOORD,-NUMCOL:NUMCOL),DIP(MCOORD,-
50     9NUMCOL:NUMCOL),TETA(MCOORD,-NUMCOL:NUMCOL),A3(MCOORD,-
51     XNUMCOL:NUMCOL),ZSOL(MSOIL,MCOORD,-NUMCOL:NUMCOL,NCORN),W3(MCOORD,
52     XNUMCOL:NUMCOL),U3(MCOORD,-NUMCOL:NUMCOL)
53     REAL FSCUM(MAXSUR)
54     REAL XAPRIN,YMIN1,DYMIN1,YMIN2,DYMIN2,DATOUT,FS,YPA1,YPA2
55     INTEGER NXPOL(MCOORD),NXSTAR(MCOORD),NXEND(MCOORD),UCOND(MSOIL)
56     INTEGER DIFF,NX,NY,NSOIL,NSURFA,NSURF,NYSTAR,NYPOL,NYEND
57 *----- Code -----
58     PRINT 20, 1, ' '
59     PRINT *, ' *****'
60     PRINT *, ' *      3D-PCSTABL : Three Dimesional Slope Stability *'
61     PRINT *, ' *                               from Purdue University *'
62     PRINT *, ' *****'
63     PRINT *, ' '
64     PRINT *, ' '
65 *.....read data .....
66     CALL INDATA (NX,NY,NSOIL,X,Y,Z,XAPRIN,YMIN1,DYMIN1,YMIN2,DYMIN2,
67     1XMIN1,DXMIN1,XMIN2,DXMIN2,ZMIN,DZMIN,GAMA,COES,FI,UCOND,RU,UZ,
68     2DATOUT)
69     WRITE (*,30) ' Enter number of surfaces to be generated : '

```

^H PCSTBL3D v2.0

 PAGE 3
 04-12-91
 12:06:11

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70      READ (*,*) NSURFA
71      WRITE (*,*) ' '
72      WRITE (*,*) ' Running ....'
73      DO 10 NSURF=1,NSURFA
74          WRITE (*,*) 'Now analysing surface #',NSURF
75      *.....generate 3D-surface.....
76          CALL GENSUR (NX,NY,X,Y,Z,XAPRIN,YMIN1,DYMIN1,YMIN2,DYMIN2,
77      1          XMIN1,DXMIN1,XMIN2,DXMIN2,ZMIN,DZMIN,YPA1,NXEND,YPA2,NYSTAR,
78      2          NYPOL,XIPOL,XFPOL,NXSTAR,NXPOL,XPOL,ZPOL,YPOL,XCOL,YCOL,ZCOL
79      3          NCOLEF,NCORIG,NPTLEF,NPTRIG,ZSOL,NSOIL,NYEND,DIFF)
80      *.....calculate parameters of columns.....
81          CALL PARAM1 (NYPOL,XCOL,YCOL,ZCOL,ZSOL,NCOLEF,NCORIG,ALPHAXZ,
82      1          ALPHAYZ,DIP,TETA,A3,GAMA,RU,W3,U3,NSOIL)
83      *.....calculate factor of safety.....
84          CALL FSCALC (COES,FI,A3,W3,U3,DIP,ALPHAYZ,FS,NYPOL,NCOLEF,
85      1          NCORIG)
86          IF (FS.GT.0.1) THEN
87              FSCUM(NFS)=FS
88              NFS=NFS+1
89          END IF
90      *.....store and print 10 most critical surfaces.....
91          CALL CRITIC (NYPOL,NXPOL,XPOL,YPOL,ZPOL,NXSTAR,NXEND,NSURF,
92      1          NSURFA,FS,NX,NY,X,Y,Z,NYSTAR,NYEND)
93      *.....end of loop.....
94      10  CONTINUE
95          CALL WEIBUL (NFS,FSCUM)
96          WRITE (*,*) ' Finished.. Use graphical processor to check surface
97          STOP
98      C
99      20  FORMAT (A,T2,A)
100     30  FORMAT (A)
101     END

```

***** THESIS.FOR(101) : warning F4999: XDIFF : variable declared but not used

main Local Symbols

Name	Class	Type	Size	Offset
Z	local	REAL*4	12500	0000
NCORIG.	local	REAL*4	100	0000
NSOIL	local	INTEGER*4	4	0002
NSURF	local	INTEGER*4	4	0006
NSURFA.	local	INTEGER*4	4	000a
DATOUT.	local	REAL*4	4	000e
NYPOL	local	INTEGER*4	4	0012
XAPRIN.	local	REAL*4	4	0016
DIFF.	local	INTEGER*4	4	001a
FS.	local	REAL*4	4	001e
YPA1.	local	REAL*4	4	0022
YPA2.	local	REAL*4	4	0026
NYSTAR.	local	INTEGER*4	4	002a
NX.	local	INTEGER*4	4	002e
NFS	local	INTEGER*4	4	0032
NY.	local	INTEGER*4	4	0036
YMIN1	local	REAL*4	4	003a
YMIN2	local	REAL*4	4	003e

^H PCSTBL3D v2.0

 PAGE 4
 04-12-91
 12:06:11

Microsoft FORTRAN Optimizing Compiler Version 5.00

main Local Symbols

Name	Class	Type	Size	Offset
DYMIN1.	local	REAL*4	4	0042
DYMIN2.	local	REAL*4	4	0046
NYEND	local	INTEGER*4	4	004a
DZMIN	local	REAL*4	2500	0064
XPOL.	local	REAL*4	2500	0a28
YPOL.	local	REAL*4	100	13ec
ZPOL.	local	REAL*4	2500	1450
U3.	local	REAL*4	2100	1e14
ZSOL.	local	REAL*4	42000	2648
DIP	local	REAL*4	2100	30d4
NXSTAR.	local	INTEGER*4	100	3908
RU.	local	REAL*4	20	396c
COES.	local	REAL*4	20	3980
XMIN1	local	REAL*4	100	3994
TETA.	local	REAL*4	2100	39f8
XMIN2	local	REAL*4	100	422c
UZ.	local	REAL*4	12500	4290
XDIFF	local	REAL*4	100	7364
DXMIN1.	local	REAL*4	100	73c8
DXMIN2.	local	REAL*4	100	742c
A3.	local	REAL*4	2100	7490
XCOL.	local	REAL*4	8400	7cc4
NCOLEF.	local	REAL*4	100	9d94
YCOL.	local	REAL*4	8400	9df8
ZCOL.	local	REAL*4	8400	bec8
NPTLEF.	local	REAL*4	100	ca58
XFPOL	local	REAL*4	100	cabc
W3.	local	REAL*4	2100	cb20
XIPOL	local	REAL*4	100	d354
FI.	local	REAL*4	20	d3b8
NXPOL	local	INTEGER*4	100	d3cc
NPTRIG.	local	REAL*4	100	d430
GAMA.	local	REAL*4	20	d494
ALPHAXZ	local	REAL*4	2100	d4a8
X	local	REAL*4	100	dcdc
ALPHAYZ	local	REAL*4	2100	dd40
UCOND	local	INTEGER*4	20	df98
NXEND	local	INTEGER*4	100	dfac
FSCUM	local	REAL*4	8000	e010
Y	local	REAL*4	100	e574
ZMIN.	local	REAL*4	2500	e5d8

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

^H PCSTBL3D v2.0

PAGE 5
04-12-91
12:06:11

Microsoft FORTRAN Optimizing Compiler Version 5.00

Global Symbols

Name	Class	Type	Size	Offset
CRITIC.	extern	***	***	***
FSCALC.	extern	***	***	***
GENSUR.	extern	***	***	***
INDATA.	extern	***	***	***
PARAM1.	extern	***	***	***
WEIBUL.	extern	***	***	***
main.	FSUBRT	***	***	0000

Code size = 04a5 (1189)
Data size = 01c1 (449)
Bss size = 004e (78)

No errors detected

^H

PAGE 1
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1 $nottruncate
2 $title:'PCSTBL3D v2.0'
3 $subtitle:'Left Side Columns'
4     SUBROUTINE COLUML (YPA1,YPA2,NYSTAR,NYPOL,X,Y,Z,XAPRIN,XIPOL,
5     1NXSTAR,NXPOL,XPOL,YPOL,XCOL,YCOL,ZCOL,NCOLEF,NPTLEF,ZPOL,ZSOL,
6     2XDIFF,NSOIL,NPPOL)
7 *
8 *===== Prologue =====
9 *
10 * Purpose:
11 *     Calculates coordinates of the columns for the left side
12 *
13 * History:
14 *     Version      Programmer      Date      Description
15 *     -----
16 *     1.0          J.E.Thomas
17 *     2.0          G.Moshkovich    created
18 *                                     code beautified,
19 *                                     variables declared
20 * IN args/commons      Units      Description
21 * -----
22 * x
23 * y
24 * z
25 *
26 * OUT args/commons     Units      Description
27 * -----
28 *
29 * INOUT args/commons   Units      Description
30 * -----
31 *
32 * Processing:
33 *     Main processing module. Gets in/output datafile names, number
34 * of critical surfaces to be generated, and invoke calculation routines.
35 *
36 * Special Requirements:
37 *     none
38 *
39 *----- Include files -----
40 $include:'CURRENTF.INC'
```

```

***** Begin listing of: CURRENTF.INC
1  *
2  *===== Prologue =====
3  *
4  * Purpose:
5  *   Include file with parameter declaration for the program PCSTL3D
6  *
7  * History:
8  *   Version      Programmer      Date      Description
9  *   -----      -
10 *   1.0          G.Moshkovich    ----      created
11 *
12 * IN args/commons      Units      Description
13 * -----
14 *
15 * OUT args/commons     Units      Description
16 * -----
17 *

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 2
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

18 * INOUT args/commons          Units          Description
19 * -----
20 *
21 * Processing:
22 *   none
23 *
24 * Special Requirements:
25 *   none
26 *
27 *----- Include files -----
28 *----- Constant declarations -----
29     integer MCOORD, MSOIL, NUMCOL, NCORN
30
31     parameter (MCOORD = 25, MSOIL = 5)
32     parameter (NUMCOL = 10, NCORN = 4)
33     PARAMETER (MAXSUR=2000)
34
35 *----- MCOORD - maximum number of points in each direction
36 *----- MSOIL - maximum number of soil layers
37 *----- NUMCOL - maximum number of generated columns
38 *----- NCORN - number of nods for each column (do not change)
39
40 *----- Argument declarations -----
41 *----- Global/External declarations -----
42 *----- Local declarations -----
43
44
45
46 ***** End listing of: CURRENTF.INC
47 *----- Constant declarations -----
48 *----- Argument declarations -----
49     REAL X(MCOORD), Y(MCOORD), Z(MSOIL, MCOORD, MCOORD), XIPOL(MCOORD),
50     1XPOL(MCOORD, MCOORD), ZPOL(MCOORD, MCOORD), YPOL(MCOORD), XCOL(MCOORD,
51     2NUMCOL: NUMCOL, NCORN), YCOL(MCOORD, -NUMCOL: NUMCOL, NCORN),
52     3ZCOL(MCOORD, -NUMCOL: NUMCOL, NCORN), ZSOL(MSOIL, MCOORD, -
53     4NUMCOL: NUMCOL, NCORN)
54     REAL YPA1, YPA2
55     INTEGER XAPRIN, XROW
56     INTEGER XDIFF(MCOORD), NXPOL(MCOORD), NXSTAR(MCOORD), NCOLEF(MCOORD)
57     1NPTLEF(MCOORD)
58 *----- Global/External declarations -----
59     EXTERNAL COORYZ
60 *----- Local declarations -----
61     REAL YTES1, XTES1, YTES2, XTES2, A, B, YTESTE, ZFINAL, ALPHA
62     INTEGER I, NI, NY, M1, NNN, J, II, NNJ, NNJ1, JJJ, JJJ1
63 *----- Code -----
64 C
65 C calculate coordinates of basis of columns
66 C
67 C
68 C PART 1 - 1st SERIES OF COLUMNS - LEFT SIDE
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 3
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70 C -----
71 C number of columns
72 C -----
73 NCOLEF(1)=(XAPRIN-NXSTAR(2)+1)
74 C -----
75 C number of points
76 C -----
77 NPTLEF(2)=NCOLEF(1)+1
78 C -----
79 C transfer coordinates
80 C -----
81 DO 40 I=1,NCOLEF(1)
82     NI=-I
83     XCOL(1,NI,1)=XPOL(NPTLEF(2)+1-I,2)
84     XCOL(1,NI,2)=XPOL(NPTLEF(2)+1-I,2)
85     XCOL(1,NI,3)=XPOL(NPTLEF(2)-I,2)
86     XCOL(1,NI,4)=XPOL(NPTLEF(2)-I,2)
87     YCOL(1,NI,1)=A*XCOL(1,NI,1)+B
88     YCOL(1,NI,2)=Y(NYSTAR)
89     YCOL(1,NI,3)=Y(NYSTAR)
90     YCOL(1,NI,4)=A*XCOL(1,NI,4)+B
91 C -----
92 C define xrow to be used
93 C -----
94     XROW=XAPRIN+1-I
95     YTESTE=YCOL(1,NI,1)
96     CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
97     ZCOL(1,NI,1)=ZFINAL
98     ZCOL(1,NI,2)=ZPOL(NPTLEF(2)+1-I,2)
99     ZCOL(1,NI,3)=ZPOL(NPTLEF(2)-I,2)
100 DO 10 M1=1,NSOIL
101     NNN=NPTLEF(2)+1-I
102     ZSOL(M1,1,NI,1)=ZCOL(1,NI,1)
103     ZSOL(M1,1,NI,2)=Z(M1,NNN+XDIFF(2),NYSTAR)
104     ZSOL(M1,1,NI,3)=Z(M1,NNN-1+XDIFF(2),NYSTAR)
105 10 CONTINUE
106     IF (XROW.EQ.NXSTAR(2)) THEN
107         ZCOL(1,NI,4)=ZPOL(1,2)
108         DO 20 M1=1,NSOIL
109             ZSOL(M1,1,NI,4)=ZCOL(1,NI,4)
110 20 CONTINUE
111     ELSE
112         YTESTE=YCOL(1,NI,4)
113         CALL COORYZ (XROW-1,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
114         ZCOL(1,NI,4)=ZFINAL
115         DO 30 M1=1,NSOIL
116             ZSOL(M1,1,NI,4)=ZCOL(1,NI,4)
117 30 CONTINUE
118     END IF
119 40 CONTINUE
120     XPOL(1,1)=X(XAPRIN)
121     ZPOL(1,1)=ZCOL(1,-1,1)
122     NXPOL(1)=1
123 C -----
124 C Part 2 - FOR ALL NEXT SERIES OF COLUMNS - LEFT SIDE
125 C ( 2nd to nypol-2 )
126 C -----
127 DO 220 J=2,NYPOL-2

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 4
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

128 C -----
129 C check the number of points of the ith and the i+1 th
130 C -----
131 C      NCOLEF(J)=(XAPRIN-NXSTAR(J)+1)
132 C      NCOLEF(J+1)=(XAPRIN-NXSTAR(J+1)+1)
133 C -----
134 C A) --> if number of points of j+1 > than j <---
135 C -----
136 C      IF (NCOLEF(J+1).GT.NCOLEF(J)) THEN
137 C -----
138 C      number of columns
139 C -----
140 C      NPTLEF(J)=NCOLEF(J)+1
141 C      NPTLEF(J+1)=NCOLEF(J+1)+1
142 C -----
143 C calculate line that connects extremities
144 C -----
145 C      YTES1=YPOL(J+1)
146 C      XTES1=XPOL(1,J+1)
147 C      YTES2=YPOL(J)
148 C      XTES2=XPOL(1,J)
149 C      A=(YTES1-YTES2)/(XTES1-XTES2)
150 C      B=YTES1-A*XTES1
151 C -----
152 C transfer coordinates
153 C -----
154 C      DO 110 II=1,NCOLEF(J+1)
155 C          NI=-II
156 C          NNJ=NPTLEF(J)+1-II
157 C          NNJ1=NPTLEF(J+1)+1-II
158 C          JJJ=NYSTAR+J-2
159 C          JJJ1=NYSTAR+J+1-2
160 C          XCOL(J,NI,1)=XPOL(NPTLEF(J+1)+1-II,J+1)
161 C          XCOL(J,NI,2)=XPOL(NPTLEF(J+1)+1-II,J+1)
162 C          XCOL(J,NI,3)=XPOL(NPTLEF(J+1)-II,J+1)
163 C          XCOL(J,NI,4)=XPOL(NPTLEF(J+1)-II,J+1)
164 C          IF (II.LE.NCOLEF(J)-1) THEN
165 C              YCOL(J,NI,1)=YPOL(J)
166 C              YCOL(J,NI,2)=YPOL(J+1)
167 C              YCOL(J,NI,3)=YPOL(J+1)
168 C              YCOL(J,NI,4)=YPOL(J)
169 C          ELSE
170 C              IF (II.EQ.NCOLEF(J)) THEN
171 C                  YCOL(J,NI,1)=YPOL(J)
172 C              ELSE
173 C                  YCOL(J,NI,1)=A*XCOL(J,NI,1)+B
174 C              END IF
175 C              YCOL(J,NI,2)=YPOL(J+1)
176 C              YCOL(J,NI,3)=YPOL(J+1)
177 C              YCOL(J,NI,4)=A*XCOL(J,NI,4)+B
178 C          END IF
179 C -----
180 C define xrow to be used
181 C -----
182 C      IF (II.LE.NCOLEF(J)-1) THEN
183 C          ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+1-II,J)
184 C          ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+1-II,J+1)
185 C          ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-II,J+1)

```


^H PCSTBL3D v2.0
Left Side Columns

PAGE 5
04-12-91
12:06:28

```

Line#  Source Line      Microsoft FORTRAN Optimizing Compiler Version 5.00

186      ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-II,J)
187      DO 50 M1=1,NSOIL
188          ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
189          ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
190          ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
191          ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)
192      50      CONTINUE
193      ELSE
194          IF (II.EQ.NCOLEF(J)) THEN
195              ZCOL(J,NI,1)=ZCOL(J,NI+1,4)
196              DO 60 M1=1,NSOIL
197                  ZSOL(M1,J,NI,1)=ZSOL(M1,J,NI+1,4)
198      60      CONTINUE
199      ELSE
200          XROW=XAPRIN+1-II
201          YTESTE=YCOL(J,NI,1)
202          CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
203          ZCOL(J,NI,1)=ZFINAL
204          DO 70 M1=1,NSOIL
205              ZSOL(M1,J,NI,1)=ZFINAL
206      70      CONTINUE
207      END IF
208      ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+1-II,J+1)
209      ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-II,J+1)
210      DO 80 M1=1,NSOIL
211          ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
212          ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
213      80      CONTINUE
214      XROW=XAPRIN+1-II
215      IF (XROW.EQ.NXSTAR(J+1)) THEN
216          ZCOL(J,NI,4)=ZPOL(1,J+1)
217          DO 90 M1=1,NSOIL
218              ZSOL(M1,J,NI,4)=ZPOL(1,J+1)
219      90      CONTINUE
220      ELSE
221          YTESTE=YCOL(J,NI,4)
222          CALL COORYZ (XROW-1,Y,Z,NY,YTESTE,ZFINAL,
223      1          ALPHA)
224          ZCOL(J,NI,4)=ZFINAL
225          DO 100 M1=1,NSOIL
226              ZSOL(M1,J,NI,4)=ZFINAL
227      100     CONTINUE
228      END IF
229      END IF
230      110     CONTINUE
231      C      -----
232      C      Here ends the condition J+1 > j
233      C      -----
234      C      END IF
235      C      -----
236      C      B) --> if number of points of j+1 = number of points of j
237      C      -----
238      C      IF (NCOLEF(J+1).EQ.NCOLEF(J)) THEN
239      C      -----
240      C      number of points
241      C      -----
242      C      NPTLEF(J)=NCOLEF(J)+1
243      C      NPTLEF(J+1)=NCOLEF(J+1)+1

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 6
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

244 C -----
245 C transfer coordinates
246 C -----
247 C      DO 130 II=1,NCOLEF(J+1)
248 C          NI=-II
249 C          NNJ=NPTLEF(J)+1-II
250 C          NNJ1=NPTLEF(J+1)+1-II
251 C          JJJ=NYSTAR+J-2
252 C          JJJ1=NYSTAR+J+1-2
253 C          XCOL(J,NI,1)=XPOL(NPTLEF(J)+1-II,J)
254 C          XCOL(J,NI,2)=XPOL(NPTLEF(J+1)+1-II,J+1)
255 C          XCOL(J,NI,3)=XPOL(NPTLEF(J+1)-II,J+1)
256 C          XCOL(J,NI,4)=XPOL(NPTLEF(J)-II,J)
257 C          YCOL(J,NI,1)=YPOL(J)
258 C          YCOL(J,NI,2)=YPOL(J+1)
259 C          YCOL(J,NI,3)=YPOL(J+1)
260 C          YCOL(J,NI,4)=YPOL(J)
261 C          ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+1-II,J)
262 C          ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+1-II,J+1)
263 C          ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-II,J+1)
264 C          ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-II,J)
265 C          DO 120 M1=1,NSOIL
266 C              ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
267 C              ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
268 C              IF (II.NE.NCOLEF(J+1)) THEN
269 C                  ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
270 C                  ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)
271 C              ELSE
272 C                  ZSOL(M1,J,NI,3)=ZCOL(J,NI,3)
273 C                  ZSOL(M1,J,NI,4)=ZCOL(J,NI,4)
274 C              END IF
275 C          CONTINUE
276 C      CONTINUE
277 C -----
278 C Here ends the condition j+1 = j
279 C -----
280 C      END IF
281 C -----
282 C A) --> if number of points of j+1 < than j <---
283 C -----
284 C      IF (NCOLEF(J+1).LT.NCOLEF(J)) THEN
285 C -----
286 C number of points
287 C -----
288 C          NPTLEF(J)=NCOLEF(J)+1
289 C          NPTLEF(J+1)=NCOLEF(J+1)+1
290 C -----
291 C calculate line that connects extremities
292 C -----
293 C          YTES1=YPOL(J+1)
294 C          XTES1=XPOL(1,J+1)
295 C          YTES2=YPOL(J)
296 C          XTES2=XPOL(1,J)
297 C          A=(YTES1-YTES2)/(XTES1-XTES2)
298 C          B=YTES1-A*XTES1
299 C -----
300 C transfer coordinates
301 C -----

```


^H PCSTBL3D v2.0
Left Side Columns

PAGE 7
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

302      DO 210 II=1,NCOLEF(J)
303          NI=-II
304          NNJ=NPTLEF(J)+1-II
305          NNJ1=NPTLEF(J+1)+1-II
306          JJJ=NYSTAR+J-2
307          JJJ1=NYSTAR+J+1-2
308          XCOL(J,NI,1)=XPOL(NPTLEF(J)+1-II,J)
309          XCOL(J,NI,2)=XPOL(NPTLEF(J)+1-II,J)
310          XCOL(J,NI,3)=XPOL(NPTLEF(J)-II,J)
311          XCOL(J,NI,4)=XPOL(NPTLEF(J)-II,J)
312          IF (II.LE.NCOLEF(J+1)-1) THEN
313              YCOL(J,NI,1)=YPOL(J)
314              YCOL(J,NI,2)=YPOL(J+1)
315              YCOL(J,NI,3)=YPOL(J+1)
316              YCOL(J,NI,4)=YPOL(J)
317          ELSE
318              YCOL(J,NI,1)=YPOL(J)
319              IF (II.EQ.NCOLEF(J+1)) THEN
320                  YCOL(J,NI,2)=YPOL(J+1)
321              ELSE
322                  YCOL(J,NI,2)=A*XCOL(J,NI,2)+B
323              END IF
324              YCOL(J,NI,3)=A*XCOL(J,NI,3)+B
325              YCOL(J,NI,4)=YPOL(J)
326          END IF
327  C -----
328  C define xrow to be used
329  C -----
330          IF (II.LE.NCOLEF(J+1)-1) THEN
331              ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+1-II,J)
332              ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+1-II,J+1)
333              ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-II,J+1)
334              ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-II,J)
335              DO 140 M1=1,NSOIL
336                  ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
337                  ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
338                  ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
339                  ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)
340 140      CONTINUE
341          ELSE
342              ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+1-II,J)
343              DO 150 M1=1,NSOIL
344                  ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
345 150      CONTINUE
346              XROW=XAPRIN+1-II
347              IF (II.EQ.NCOLEF(J+1)) THEN
348                  ZCOL(J,NI,2)=ZCOL(J,NI+1,3)
349                  DO 160 M1=1,NSOIL
350                      ZSOL(M1,J,NI,2)=ZSOL(M1,J,NI+1,3)
351 160      CONTINUE
352              ELSE
353                  YTESTE=YCOL(J,NI,2)
354                  CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
355                  ZCOL(J,NI,2)=ZFINAL
356                  DO 170 M1=1,NSOIL
357                      ZSOL(M1,J,NI,2)=ZFINAL
358 170      CONTINUE
359          END IF

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 8
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

360      IF (XROW.EQ.NXSTAR(J)) THEN
361          ZCOL(J,NI,3)=ZPOL(1,J)
362          DO 180 M1=1,NSOIL
363              ZSOL(M1,J,NI,3)=ZPOL(1,J)
364      180      CONTINUE
365      ELSE
366          YTESTE=YCOL(J,NI,3)
367          CALL COORYZ (XROW-1,Y,Z,NY,YTESTE,ZFINAL,
368      1          ALPHA)
369          ZCOL(J,NI,3)=ZFINAL
370          DO 190 M1=1,NSOIL
371              ZSOL(M1,J,NI,3)=ZFINAL
372      190      CONTINUE
373      END IF
374      END IF
375      ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-II,J)
376      DO 200 M1=1,NSOIL
377          ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)
378      200      CONTINUE
379      210      CONTINUE
380      C -----
381      C Here ends the condition J+1 < j
382      C -----
383      END IF
384      C -----
385      C This label 3 ends the process for the left side
386      C -----
387      220 CONTINUE
388      C -----
389      C PART 3 - LAST SERIES OF COLUMNS - LEFT SIDE
390      C -----
391      YTES1=YPA2
392      XTES1=X(XAPRIN)
393      YTES2=YPOL(NYPOL-1)
394      XTES2=XPOL(1,NYPOL-1)
395      A=(YTES1-YTES2)/(XTES1-XTES2)
396      B=YTES1-A*XTES1
397      C -----
398      C number of columns
399      C -----
400      NCOLEF(NYPOL-1)=(XAPRIN-NXSTAR(NYPOL-1)+1)
401      C -----
402      C number of points
403      C -----
404      NPTLEF(NYPOL-1)=NCOLEF(NYPOL-1)+1
405      NPPOL=NYPOL-1
406      C -----
407      C transfer coordinates
408      C -----
409      DO 280 I=1,NCOLEF(NPPOL)
410          NI=-I
411          NNJ=NPTLEF(NPPOL)+1-I
412          JJJ=NYSTAR+NPPOL-2
413          XCOL(NPPOL,NI,1)=XPOL(NPTLEF(NPPOL)+1-I,NPPOL)
414          XCOL(NPPOL,NI,2)=XPOL(NPTLEF(NPPOL)+1-I,NPPOL)
415          XCOL(NPPOL,NI,3)=XPOL(NPTLEF(NPPOL)-I,NPPOL)
416          XCOL(NPPOL,NI,4)=XPOL(NPTLEF(NPPOL)-I,NPPOL)
417          YCOL(NPPOL,NI,1)=YPOL(NPPOL)

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 9
04-12-91
12:06:28

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

418          YCOL(NPPOL,NI,2)=A*XCOL(NPPOL,NI,2)+B
419          YCOL(NPPOL,NI,3)=A*XCOL(NPPOL,NI,3)+B
420          YCOL(NPPOL,NI,4)=YPOL(NPPOL)
421 C -----
422 C define xrow to be used
423 C -----
424          ZCOL(NPPOL,NI,1)=ZPOL(NPTLEF(NPPOL)+1-I,NPPOL)
425          DO 230 M1=1,NSOIL
426              ZSOL(M1,NPPOL,NI,1)=Z(M1,NNJ+XDIFF(NPPOL),JJJ)
427 230      CONTINUE
428          XROW=XAPRIN+1-I
429          YTESTE=YCOL(NPPOL,NI,2)
430          CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
431          ZCOL(NPPOL,NI,2)=ZFINAL
432          DO 240 M1=1,NSOIL
433              ZSOL(M1,NPPOL,NI,2)=ZFINAL
434 240      CONTINUE
435          IF (XROW.EQ.NXSTAR(NPPOL)) THEN
436              ZCOL(NPPOL,NI,3)=ZPOL(1,NPPOL)
437              DO 250 M1=1,NSOIL
438                  ZSOL(M1,NPPOL,NI,3)=ZPOL(1,NPPOL)
439 250          CONTINUE
440          ELSE
441              YTESTE=YCOL(NPPOL,NI,3)
442              CALL COORYZ (XROW-1,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
443              ZCOL(NPPOL,NI,3)=ZFINAL
444              DO 260 M1=1,NSOIL
445                  ZSOL(M1,NPPOL,NI,3)=ZFINAL
446 260          CONTINUE
447          END IF
448          ZCOL(NPPOL,NI,4)=ZPOL(NPTLEF(NPPOL)-I,NPPOL)
449          DO 270 M1=1,NSOIL
450              ZSOL(M1,NPPOL,NI,4)=Z(M1,NNJ-1+XDIFF(NPPOL),JJJ)
451 270          CONTINUE
452 280      CONTINUE
453 C -----
454 C The left side calculations have finished
455 C -----
456          RETURN
457          END

```

COLUML Local Symbols

Name	Class	Type	Size	Offset
NPPOL	param			0006
NSOIL	param			000a
XDIFF	param			000e
ZSOL	param			0012
ZPOL	param			0016
NPTLEF	param			001a
NCOLEF	param			001e
ZCOL	param			0022
YCOL	param			0026
XCOL	param			002a
YPOL	param			002e
XPOL	param			0032

^H PCSTBL3D v2.0
Left Side Columns

PAGE 10
04-12-91
12:06:28

Microsoft FORTRAN Optimizing Compiler Version 5.00

COLUML Local Symbols

Name	Class	Type	Size	Offset
NXPOL	param			0036
NXSTAR.	param			003a
XIPOL	param			003e
XAPRIN.	param			0042
Z	param			0046
Y	param			004a
X	param			004e
NYPOL	param			0052
NYSTAR.	param			0056
YPA2.	param			005a
YPA1.	param			005e
A	local	REAL*4	4	0000
B	local	REAL*4	4	0004
ZFINAL.	local	REAL*4	4	0008
I	local	INTEGER*4	4	000c
J	local	INTEGER*4	4	0010
JJJ1.	local	INTEGER*4	4	0014
XROW.	local	INTEGER*4	4	0018
II.	local	INTEGER*4	4	001c
NNJ1.	local	INTEGER*4	4	0020
NI.	local	INTEGER*4	4	0024
JJJ	local	INTEGER*4	4	0028
YTESTE.	local	REAL*4	4	002c
NNJ	local	INTEGER*4	4	0030
ALPHA	local	REAL*4	4	0034
NY.	local	INTEGER*4	4	0038
NNN	local	INTEGER*4	4	003c
XTES1	local	REAL*4	4	0040
XTES2	local	REAL*4	4	0044
YTES1	local	REAL*4	4	0048
YTES2	local	REAL*4	4	004c
M1.	local	INTEGER*4	4	0050

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
COLUML.	FSUBRT	***	***	0000
COORYZ.	extern	***	***	***

Code size = 2e5e (11870)
Data size = 0000 (0)

^H PCSTBL3D v2.0
Left Side Columns

PAGE 11
04-12-91
12:06:28

Microsoft FORTRAN Optimizing Compiler Version 5.00

Bss size = 0054 (84)

No errors detected

^H

PAGE 1
04-12-91
12:06:50

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1 $title:'PCSTBL3D v2.0'
2 $subtitle:'Left Side Columns'
3 C This subroutine calculates coordinates of columns for the right side
4 C-----
5      SUBROUTINE COLUMR (YPA1,YPA2,NYSTAR,NYPOL,NPPOL,X,Y,Z,XAPRIN,
6        1XFPOL,NXSTAR,NXPOL,XPOL,YPOL,ZPOL,XCOL,YCOL,ZCOL,NCOLEF,NCORIG,
7        2NPTLEF,NPTRIG,NXEND,ZSOL,XDIFF,NSOIL)
8 $include:'CURRENTF.INC'
9 ***** Begin listing of: CURRENTF.INC
10 *
11 *
12 *===== Prologue =====
13 *
14 * Purpose:
15 *   Include file with parameter declaration for the program PCSTL3D
16 *
17 * History:
18 *   Version      Programmer      Date      Description
19 *   -----
20 *   1.0          G.Moshkovich      ----      created
21 *
22 * IN args/commons      Units      Description
23 * -----
24 *
25 * OUT args/commons      Units      Description
26 * -----
27 *
28 * INOUT args/commons      Units      Description
29 * -----
30 *
31 * Processing:
32 *   none
33 *
34 * Special Requirements:
35 *   none
36 *
37 *----- Include files -----
38 *----- Constant declarations -----
39      integer MCOORD, MSOIL, NUMCOL, NCORN
40
41      parameter (MCOORD = 25, MSOIL = 5)
42      parameter (NUMCOL = 10, NCORN = 4)
43      PARAMETER (MAXSUR=2000)
44
45 *----- MCOORD - maximum number of points in each direction
46 *----- MSOIL - maximum number of soil layers
47 *----- NUMCOL - maximum number of generated columns
48 *----- NCORN - number of nodes for each column (do not change)
49
50 *----- Argument declarations -----
51 *----- Global/External declarations -----
52 *----- Local declarations -----
53
54 ***** End listing of: CURRENTF.INC
55      DIMENSION X(MCOORD), Y(MCOORD), Z(MSOIL,MCOORD,MCOORD),
56      1XFPOL(MCOORD), NXPOL(MCOORD), NXSTAR(MCOORD), NXEND(MCOORD),
57      2XPOL(MCOORD,MCOORD), ZPOL(MCOORD,MCOORD), YPOL(MCOORD),

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 2
04-12-91
12:06:50

```

Line#  Source Line      Microsoft FORTRAN Optimizing Compiler Version 5.00

12      3XCOL(MCOORD,-NUMCOL:NUMCOL,NCORN), YCOL(MCOORD,-NUMCOL:NUMCOL,
13      4NCORN), ZCOL(MCOORD,-NUMCOL:NUMCOL,NCORN), NCOLEF(MCOORD),
14      5NPTLEF(MCOORD), NCORIG(MCOORD), NPTRIG(MCOORD), ZSOL(MSOIL,MCOORD
15      6-NUMCOL:NUMCOL,NCORN)
16      INTEGER XAPRIN,XROW,XDIFF(MCOORD)
17      XPOL(1,NYPOL)=X(XAPRIN)
18      ZPOL(1,NYPOL)=ZCOL(NPPOL,-1,2)
19      NXPOL(NYPOL)=1
20      C -----
21      C PART 4 - 1st SERIES OF COLUMNS - RIGHT SIDE
22      C -----
23      YTES1=YPA1
24      XTES1=X(XAPRIN)
25      YTES2=Y(NYSTAR)
26      XTES2=XFPOL(2)
27      A=(YTES1-YTES2)/(XTES1-XTES2)
28      B=YTES1-A*XTES1
29      C -----
30      C number of columns
31      C -----
32      NCORIG(1)=NXEND(2)-XAPRIN+1
33      C -----
34      C number of points
35      C -----
36      NPTLEF(2)=(XAPRIN-NXSTAR(2)+2)
37      NPTRIG(2)=NCORIG(1)+1
38      C -----
39      C transfer coordinates
40      C -----
41      DO 30 I=1,NCORIG(1)
42          NI=I
43          XCOL(1,NI,1)=XPOL(NPTLEF(2)+I,2)
44          XCOL(1,NI,2)=XPOL(NPTLEF(2)+I,2)
45          XCOL(1,NI,3)=XPOL(NPTLEF(2)-1+I,2)
46          XCOL(1,NI,4)=XPOL(NPTLEF(2)-1+I,2)
47          YCOL(1,NI,1)=A*XCOL(1,NI,1)+B
48          YCOL(1,NI,2)=Y(NYSTAR)
49          YCOL(1,NI,3)=Y(NYSTAR)
50          YCOL(1,NI,4)=A*XCOL(1,NI,4)+B
51      C -----
52      C define xrow to be used
53      C -----
54          XROW=XAPRIN+I
55          YTESTE=YCOL(1,NI,1)
56          CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
57          ZCOL(1,NI,1)=ZFINAL
58          ZCOL(1,NI,2)=ZPOL(NPTLEF(2)+I,2)
59          ZCOL(1,NI,3)=ZPOL(NPTLEF(2)-1+I,2)
60          DO 10 M1=1,NSOIL
61              NNN=NPTLEF(2)+I
62              ZSOL(M1,1,NI,1)=ZCOL(1,NI,1)
63              ZSOL(M1,1,NI,2)=Z(M1,NNN+XDIFF(2),NYSTAR)
64              ZSOL(M1,1,NI,3)=Z(M1,NNN-1+XDIFF(2),NYSTAR)
65      10 CONTINUE
66          YTESTE=YCOL(1,NI,4)
67          CALL COORYZ (XROW-1,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
68          ZCOL(1,NI,4)=ZFINAL
69          DO 20 M1=1,NSOIL

```


^H PCSTBL3D v2.0
Left Side Columns

PAGE 3
04-12-91
12:06:50

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70          ZSOL(M1,1,NI,4)=ZCOL(1,NI,4)
71 20      CONTINUE
72 30      CONTINUE
73 C      -----
74 C      Part 5 - FOR ALL NEXT SERIES OF COLUMNS - RIGHT SIDE
75 C          ( 2nd to nypol-2 )
76 C      -----
77      DO 210 J=2,NYPOL-2
78          NCOLEF(J)=(XAPRIN-NXSTAR(J)+1)
79          NCOLEF(J+1)=(XAPRIN-NXSTAR(J+1)+1)
80          NPTLEF(J)=(XAPRIN-NXSTAR(J)+2)
81          NPTLEF(J+1)=(XAPRIN-NXSTAR(J+1)+2)
82 C      -----
83 C      check the number of points of the ith and the i+1 th
84 C      -----
85          NCORIG(J)=-(XAPRIN-NXEND(J))+1
86          NCORIG(J+1)=-(XAPRIN-NXEND(J+1))+1
87 C      -----
88 C      A) --> if number of points of j+1 > than j <---
89 C      -----
90          IF (NCORIG(J+1).GT.NCORIG(J)) THEN
91 C      -----
92 C      number of POINTS
93 C      -----
94          NPTRIG(J)=NCORIG(J)+1
95          NPTRIG(J+1)=NCORIG(J+1)+1
96 C      -----
97 C      calculate line that connects extremities
98 C      -----
99          YTES1=YPOL(J+1)
100         XTES1=XPOL(NXPOL(J+1),J+1)
101         YTES2=YPOL(J)
102         XTES2=XPOL(NXPOL(J),J)
103         A=(YTES1-YTES2)/(XTES1-XTES2)
104         B=YTES1-A*XTES1
105 C      -----
106 C      transfer coordinates
107 C      -----
108         DO 100 II=1,NCORIG(J+1)
109             NI=II
110             NNJ=NPTLEF(J)+II
111             NNJ1=NPTLEF(J+1)+II
112             JJJ=NYSTAR+J-2
113             JJJ1=NYSTAR+J+1-2
114             XCOL(J,NI,1)=XPOL(NPTLEF(J+1)+II,J+1)
115             XCOL(J,NI,2)=XPOL(NPTLEF(J+1)+II,J+1)
116             XCOL(J,NI,3)=XPOL(NPTLEF(J+1)-1+II,J+1)
117             XCOL(J,NI,4)=XPOL(NPTLEF(J+1)-1+II,J+1)
118             IF (II.LE.NCORIG(J)-1) THEN
119                 YCOL(J,NI,1)=YPOL(J)
120                 YCOL(J,NI,2)=YPOL(J+1)
121                 YCOL(J,NI,3)=YPOL(J+1)
122                 YCOL(J,NI,4)=YPOL(J)
123             ELSE
124                 YCOL(J,NI,1)=A*XCOL(J,NI,1)+B
125                 YCOL(J,NI,2)=YPOL(J+1)
126                 YCOL(J,NI,3)=YPOL(J+1)
127                 IF (II.EQ.NCORIG(J)) THEN

```


^H PCSTBL3D v2.0
Left Side Columns

PAGE 4
04-12-91
12:06:50

```

Line#  Source Line      Microsoft FORTRAN Optimizing Compiler Version 5.00

128          YCOL(J,NI,4)=YPOL(J)
129          ELSE
130          YCOL(J,NI,4)=A*XCOL(J,NI,4)+B
131          END IF
132          END IF
133  C      -----
134  C      define xrow to be used
135  C      -----
136          IF (II.LE.NCORIG(J)-1) THEN
137          ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+II,J)
138          ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+II,J+1)
139          ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-1+II,J+1)
140          ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-1+II,J)
141          DO 40 M1=1,NSOIL
142          ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
143          ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
144          ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
145          ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)
146  40      CONTINUE
147          ELSE
148          IF (II.EQ.NCORIG(J)) THEN
149          ZCOL(J,NI,4)=ZCOL(J,NI-1,1)
150          DO 50 M1=1,NSOIL
151          ZSOL(M1,J,NI,4)=ZSOL(M1,J,NI-1,1)
152  50      CONTINUE
153          ELSE
154          XROW=XAPRIN-1+II
155          YTESTE=YCOL(J,NI,4)
156          CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
157          ZCOL(J,NI,4)=ZFINAL
158          DO 60 M1=1,NSOIL
159          ZSOL(M1,J,NI,4)=ZFINAL
160  60      CONTINUE
161          END IF
162          ZCOL(J,NI,2)=ZPOL(NPTRIG(J+1)+II,J+1)
163          ZCOL(J,NI,3)=ZPOL(NPTRIG(J+1)-1+II,J+1)
164          DO 70 M1=1,NSOIL
165          ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
166          ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
167  70      CONTINUE
168          XROW=XAPRIN-1+II
169          IF (XROW.EQ.NXEND(J+1)) THEN
170          ZCOL(J,NI,1)=ZPOL(NXPOL(J+1),J+1)
171          DO 80 M1=1,NSOIL
172          ZSOL(M1,J,NI,1)=ZPOL(NXPOL(J+1),J+1)
173  80      CONTINUE
174          ELSE
175          YTESTE=YCOL(J,NI,1)
176          CALL COORYZ (XROW+1,Y,Z,NY,YTESTE,ZFINAL,
177  1      ALPHA)
178          ZCOL(J,NI,1)=ZFINAL
179          DO 90 M1=1,NSOIL
180          ZSOL(M1,J,NI,1)=ZFINAL
181  90      CONTINUE
182          END IF
183          END IF
184  100      CONTINUE
185          NCORIG(J)=- (XAPRIN-NXEND(J+1))+1

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 5
04-12-91
12:06:50

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

186          NCOLEF(J)=XAPRIN-NXSTAR(J+1)+1
187          GO TO 210
188 C      -----
189 C      Here ends the condition J+1 > j
190 C      -----
191          END IF
192 C      -----
193 C      B) --> if number of points of j+1 = number of points of j
194 C      -----
195          IF (NCORIG(J+1).EQ.NCORIG(J)) THEN
196 C      -----
197 C      number of points
198 C      -----
199          NPTRIG(J)=NCORIG(J)+1
200          NPTRIG(J+1)=NCORIG(J+1)+1
201 C      -----
202 C      transfer coordinates
203 C      -----
204          DO 120 II=1,NCORIG(J+1)
205             NI=II
206             NNJ=NPTLEF(J)+II
207             NNJ1=NPTLEF(J+1)+II
208             JJJ=NYSTAR+J-2
209             JJJ1=NYSTAR+J+1-2
210             XCOL(J,NI,1)=XPOL(NPTLEF(J)+II,J)
211             XCOL(J,NI,2)=XPOL(NPTLEF(J+1)+II,J+1)
212             XCOL(J,NI,3)=XPOL(NPTLEF(J+1)-1+II,J+1)
213             XCOL(J,NI,4)=XPOL(NPTLEF(J)-1+II,J)
214             YCOL(J,NI,1)=YPOL(J)
215             YCOL(J,NI,2)=YPOL(J+1)
216             YCOL(J,NI,3)=YPOL(J+1)
217             YCOL(J,NI,4)=YPOL(J)
218             ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+II,J)
219             ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+II,J+1)
220             ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-1+II,J+1)
221             ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-1+II,J)
222             DO 110 M1=1,NSOIL
223                IF (II.NE.NCORIG(J+1)) THEN
224                   ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIF(J),JJJ)
225                   ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIF(J+1),JJJ1)
226                ELSE
227                   ZSOL(M1,J,NI,1)=ZCOL(J,NI,1)
228                   ZSOL(M1,J,NI,2)=ZCOL(J,NI,2)
229                END IF
230                ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIF(J+1),JJJ1)
231                ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIF(J),JJJ)
232 110          CONTINUE
233 120          CONTINUE
234 C      -----
235 C      Here ends the condition j+1 = j
236 C      -----
237          END IF
238 C      -----
239 C      A) --> if number of points of j+1 < than j <---
240 C      -----
241          IF (NCORIG(J+1).LT.NCORIG(J)) THEN
242 C      -----
243 C      number of points

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 6
04-12-91
12:06:50

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

244 C -----
245      NPTRIG(J)=NCORIG(J)+1
246      NPTRIG(J+1)=NCORIG(J+1)+1
247 C -----
248 C calculate line that connects extremities
249 C -----
250      YTES1=YPOL(J+1)
251      XTES1=XPOL(NXPOL(J+1),J+1)
252      YTES2=YPOL(J)
253      XTES2=XPOL(NXPOL(J),J)
254      A=(YTES1-YTES2)/(XTES1-XTES2)
255      B=YTES1-A*XTES1
256 C -----
257 C transfer coordinates
258 C -----
259      DO 200 II=1,NCORIG(J)
260         NI=II
261         NNJ=NPTLEF(J)+II
262         NNJ1=NPTLEF(J+1)+II
263         JJJ=NYSTAR+J-2
264         JJJ1=NYSTAR+J+1-2
265         XCOL(J,NI,1)=XPOL(NPTLEF(J)+II,J)
266         XCOL(J,NI,2)=XPOL(NPTLEF(J)+II,J)
267         XCOL(J,NI,3)=XPOL(NPTLEF(J)-1+II,J)
268         XCOL(J,NI,4)=XPOL(NPTLEF(J)-1+II,J)
269         IF (II.LE.NCORIG(J+1)-1) THEN
270            YCOL(J,NI,1)=YPOL(J)
271            YCOL(J,NI,2)=YPOL(J+1)
272            YCOL(J,NI,3)=YPOL(J+1)
273            YCOL(J,NI,4)=YPOL(J)
274         ELSE
275            YCOL(J,NI,4)=YPOL(J)
276            IF (II.EQ.NCORIG(J+1)) THEN
277               YCOL(J,NI,3)=YPOL(J+1)
278            ELSE
279               YCOL(J,NI,3)=A*XCOL(J,NI,3)+B
280            END IF
281            YCOL(J,NI,2)=A*XCOL(J,NI,2)+B
282            YCOL(J,NI,1)=YPOL(J)
283         END IF
284 C -----
285 C define xrow to be used
286 C -----
287      IF (II.LE.NCORIG(J+1)-1) THEN
288         ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+II,J)
289         ZCOL(J,NI,2)=ZPOL(NPTLEF(J+1)+II,J+1)
290         ZCOL(J,NI,3)=ZPOL(NPTLEF(J+1)-1+II,J+1)
291         ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-1+II,J)
292         DO 130 M1=1,NSOIL
293            ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
294            ZSOL(M1,J,NI,2)=Z(M1,NNJ1+XDIFF(J+1),JJJ1)
295            ZSOL(M1,J,NI,3)=Z(M1,NNJ1-1+XDIFF(J+1),JJJ1)
296            ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)
297      130 CONTINUE
298      ELSE
299         ZCOL(J,NI,4)=ZPOL(NPTLEF(J)-1+II,J)
300         DO 140 M1=1,NSOIL
301            ZSOL(M1,J,NI,4)=Z(M1,NNJ-1+XDIFF(J),JJJ)

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 7
04-12-91
12:06:50

Microsoft FORTRAN Optimizing Compiler Version 5.00

Line#	Source Line	
302	140	CONTINUE
303		XROW=XAPRIN-1+II
304		IF (II.EQ.NCORIG(J+1)) THEN
305		ZCOL(J,NI,3)=ZCOL(J,NI-1,2)
306		DO 150 M1=1,NSOIL
307		ZSOL(M1,J,NI,3)=ZSOL(M1,J,NI-1,2)
308	150	CONTINUE
309		ELSE
310		YTESTE=YCOL(J,NI,3)
311		CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
312		ZCOL(J,NI,3)=ZFINAL
313		DO 160 M1=1,NSOIL
314		ZSOL(M1,J,NI,3)=ZFINAL
315	160	CONTINUE
316		END IF
317		IF (XROW.EQ.NXEND(J)) THEN
318		ZCOL(J,NI,2)=ZPOL(NXPOL(J),J)
319		DO 170 M1=1,NSOIL
320		ZSOL(M1,J,NI,2)=ZPOL(NXPOL(J),J)
321	170	CONTINUE
322		ELSE
323		YTESTE=YCOL(J,NI,2)
324		CALL COORYZ (XROW+1,Y,Z,NY,YTESTE,ZFINAL,
325	1	ALPHA)
326		ZCOL(J,NI,2)=ZFINAL
327		DO 180 M1=1,NSOIL
328		ZSOL(M1,J,NI,2)=ZFINAL
329	180	CONTINUE
330		END IF
331		END IF
332		ZCOL(J,NI,1)=ZPOL(NPTLEF(J)+II,J)
333		DO 190 M1=1,NSOIL
334		ZSOL(M1,J,NI,1)=Z(M1,NNJ+XDIFF(J),JJJ)
335	190	CONTINUE
336	200	CONTINUE
337	C	-----
338	C	Here ends the condition J+1 < j
339	C	-----
340		END IF
341	C	-----
342	C	This label 9 ends the process for the right side
343	C	-----
344	210	CONTINUE
345	C	-----
346	C	PART 6 - LAST SERIES OF COLUMNS - RIGHT SIDE
347	C	-----
348		YTES1=YPA2
349		XTES1=X(XAPRIN)
350		YTES2=YPOL(NYPOL-1)
351		XTES2=XPOL(NXPOL(NYPOL-1),NYPOL-1)
352		A=(YTES1-YTES2)/(XTES1-XTES2)
353		B=YTES1-A*XTES1
354	C	-----
355	C	number of columns
356	C	-----
357		NCORIG(NYPOL-1)=-(XAPRIN-NXEND(NYPOL-1))+1
358	C	-----
359	C	number of points

^H PCSTBL3D v2.0
Left Side Columns

PAGE 8
04-12-91
12:06:50

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

360 C -----
361 NPTRIG(NYPOL-1)=NCORIG(NYPOL-1)+1
362 NPPOL=NYPOL-1
363 NPTLEF(NPPOL)=(XAPRIN-NXSTAR(NPPOL)+2)
364 C -----
365 C transfer coordinates
366 C -----
367 DO 270 I=1,NCORIG(NPPOL)
368     NI=I
369     NNJ=NPTLEF(NPPOL)+I
370     JJJ=NYSTAR+NPPOL-2
371     XCOL(NPPOL,NI,1)=XPOL(NPTLEF(NPPOL)+I,NPPOL)
372     XCOL(NPPOL,NI,2)=XPOL(NPTLEF(NPPOL)+I,NPPOL)
373     XCOL(NPPOL,NI,3)=XPOL(NPTLEF(NPPOL)-1+I,NPPOL)
374     XCOL(NPPOL,NI,4)=XPOL(NPTLEF(NPPOL)-1+I,NPPOL)
375     YCOL(NPPOL,NI,1)=YPOL(NPPOL)
376     YCOL(NPPOL,NI,2)=A*XCOL(NPPOL,NI,2)+B
377     YCOL(NPPOL,NI,3)=A*XCOL(NPPOL,NI,3)+B
378     YCOL(NPPOL,NI,4)=YPOL(NPPOL)
379 C -----
380 C define xrow to be used
381 C -----
382     ZCOL(NPPOL,NI,4)=ZPOL(NPTLEF(NPPOL)+I-1,NPPOL)
383     DO 220 M1=1,NSOIL
384         ZSOL(M1,NPPOL,NI,4)=Z(M1,NNJ-1+XDIFF(NPPOL),JJJ)
385 220 CONTINUE
386     XROW=XAPRIN+I-1
387     YTESTE=YCOL(NPPOL,NI,3)
388     CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
389     ZCOL(NPPOL,NI,3)=ZFINAL
390     DO 230 M1=1,NSOIL
391         ZSOL(M1,NPPOL,NI,3)=ZFINAL
392 230 CONTINUE
393     IF (XROW.EQ.NXEND(NPPOL)) THEN
394         ZCOL(NPPOL,NI,2)=ZPOL(NXPOL(NPPOL),NPPOL)
395         DO 240 M1=1,NSOIL
396             ZSOL(M1,NPPOL,NI,2)=ZPOL(NXPOL(NPPOL),NPPOL)
397 240 CONTINUE
398     ELSE
399         YTESTE=YCOL(NPPOL,NI,2)
400         CALL COORYZ (XROW+1,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
401         ZCOL(NPPOL,NI,2)=ZFINAL
402         DO 250 M1=1,NSOIL
403             ZSOL(M1,NPPOL,NI,2)=ZFINAL
404 250 CONTINUE
405     END IF
406     ZCOL(NPPOL,NI,1)=ZPOL(NPTLEF(NPPOL)+I,NPPOL)
407     DO 260 M1=1,NSOIL
408         ZSOL(M1,NPPOL,NI,1)=Z(M1,NNJ+XDIFF(NPPOL),JJJ)
409 260 CONTINUE
410 270 CONTINUE
411 RETURN
412 END

```

^H PCSTBL3D v2.0
Left Side Columns

PAGE 9
04-12-91
12:06:50

Microsoft FORTRAN Optimizing Compiler Version 5.00

COLUML Local Symbols

Name	Class	Type	Size	Offset
NSOIL	param			0006
XDIFF	param			000a
ZSOL.	param			000e
NXEND	param			0012
NPTRIG.	param			0016
NPTLEF.	param			001a
NCORIG.	param			001e
NCOLEF.	param			0022
ZCOL.	param			0026
YCOL.	param			002a
XCOL.	param			002e
ZPOL.	param			0032
YPOL.	param			0036
XPOL.	param			003a
NXPOL	param			003e
NXSTAR.	param			0042
XPOL.	param			0046
XAPRIN.	param			004a
Z	param			004e
Y	param			0052
X	param			0056
NPPOL	param			005a
NYPOL	param			005e
NYSTAR.	param			0062
YPA2.	param			0066
YPA1.	param			006a
A	local	REAL*4	4	0000
B	local	REAL*4	4	0004
ZFINAL.	local	REAL*4	4	0008
I	local	INTEGER*4	4	000c
J	local	INTEGER*4	4	0010
JJJ1.	local	INTEGER*4	4	0014
XROW.	local	INTEGER*4	4	0018
II.	local	INTEGER*4	4	001c
NNJ1.	local	INTEGER*4	4	0020
NI.	local	INTEGER*4	4	0024
JJJ	local	INTEGER*4	4	0028
YTESTE.	local	REAL*4	4	002c
NNJ	local	INTEGER*4	4	0030
ALPHA	local	REAL*4	4	0034
NY.	local	INTEGER*4	4	0038
NNN	local	INTEGER*4	4	003c
XTES1	local	REAL*4	4	0040
XTES2	local	REAL*4	4	0044
YTES1	local	REAL*4	4	0048
YTES2	local	REAL*4	4	004c
MI.	local	INTEGER*4	4	0050

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10

^H PCSTBL3D v2.0
Left Side Columns

PAGE 10
04-12-91
12:06:50

Microsoft FORTRAN Optimizing Compiler Version 5.00

Symbolic Constant	Type	Value
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
COLUML.	FSUBRT	***	***	0000
COORYZ.	extern	***	***	***

Code size = 2e94 (11924)
Data size = 0000 (0)
Bss size = 0054 (84)

No errors detected

^H

PAGE 1
04-12-91
12:07:05

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE COORXZ (YFIXED,X,Z,NVERT,XTESTE,ZTESTE)
2  C      {-----input-----}{---output---}
3  $include: 'CURRENTF.INC'
**** Begin listing of: CURRENTF.INC
1  *
2  *===== Prologue =====
3  *
4  * Purpose:
5  *   Include file with parameter declaration for the program PCSTL3D
6  *
7  * History:
8  *   Version      Programmer      Date      Description
9  *   -----
10 *   1.0          G.Moshkovich      created
11 *
12 * IN args/commons      Units      Description
13 * -----
14 *
15 * OUT args/commons      Units      Description
16 * -----
17 *
18 * INOUT args/commons      Units      Description
19 * -----
20 *
21 * Processing:
22 *   none
23 *
24 * Special Requirements:
25 *   none
26 *
27 *----- Include files -----
28 *----- Constant declarations -----
29      integer MCOORD, MSOIL, NUMCOL, NCORN
30
31      parameter (MCOORD = 25, MSOIL = 5)
32      parameter (NUMCOL = 10, NCORN = 4)
33      PARAMETER (MAXSUR=2000)
34
35 *----- MCOORD - maximum number of points in each direction
36 *----- MSOIL - maximum number of soil layers
37 *----- NUMCOL - maximum number of generated columns
38 *----- NCORN - number of nods for each column (do not change)
39
40 *----- Argument declarations -----
41 *----- Global/External declarations -----
42 *----- Local declarations -----
43
44
45
**** End listing of: CURRENTF.INC
4  C.....this routine calculates the "z" coordinate of a given xteste...
5  C.....on the surface of the ground.....
6  C.....the value of y is fixed and equals yfixed.....
7      INTEGER YFIXED,YY
8      REAL X(MCOORD),Z(MSOIL,MCOORD,MCOORD)
9      YY=YFIXED
10     DO 10 J=1,NVERT
11         IF (X(J).GT.YTESTE) THEN

```


^H

PAGE 2
04-12-91
12:07:05

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

12      ALPHA=(Z(1,J,YY)-Z(1,J-1,YY))/(X(J)-X(J-1))
13      ZTESTE=Z(1,J-1,YY)+ALPHA*(XTESTE-X(J-1))
14      RETURN
15      ELSE
16      IF (X(J).EQ.XTESTE) THEN
17          ZTESTE=Z(1,J,YY)
18          ALPHA=(Z(1,J+1,YY)-Z(1,J,YY))/(X(J+1)-X(J))
19          RETURN
20      ELSE
21      END IF
22  END IF
23  10  CONTINUE
24  END

```

***** COORXZ.FOR(24) : warning F4998: YTESTE : variable used but not defined

COORXZ Local Symbols

Name	Class	Type	Size	Offset
ZTESTE.	param			0006
XTESTE.	param			000a
NVERT	param			000e
Z	param			0012
X	param			0016
YFIXED.	param			001a
J	local	INTEGER*4	4	0000
YTESTE.	local	REAL*4	4	0004
ALPHA	local	REAL*4	4	0008
YY.	local	INTEGER*4	4	000c

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
COORXZ.	FSUBRT	***	***	0000

Code size = 018d (397)
Data size = 0000 (0)
Bss size = 0010 (16)

No errors detected

^H

PAGE 1
04-12-91
12:07:11

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE COORYZ (XFIXED,Y,Z,NVERT,YTESTE,ZTESTE,ALPHA)
2      C      {-----input-----}{---output---}
3      $include:'CURRENTF.INC'
4      ***** Begin listing of: CURRENTF.INC
5      *
6      *===== Prologue =====
7      *
8      * Purpose:
9      *      Include file with parameter declaration for the program PCSTL3D
10     *
11     * History:
12     *      Version          Programmer      Date          Description
13     *      -----          -
14     *      1.0              G.Moshkovich    -----
15     *
16     * IN args/commons      Units          Description
17     * -----
18     * OUT args/commons     Units          Description
19     * -----
20     * INOUT args/commons   Units          Description
21     * -----
22     * Processing:
23     *      none
24     *
25     * Special Requirements:
26     *      none
27     *
28     *----- Include files -----
29     *----- Constant declarations -----
30     integer MCOORD, MSOIL, NUMCOL, NCORN
31
32     parameter (MCOORD = 25, MSOIL = 5)
33     parameter (NUMCOL = 10, NCORN = 4)
34     PARAMETER (MAXSUR=2000)
35
36     *----- MCOORD - maximum number of points in each direction
37     *----- MSOIL - maximum number of soil layers
38     *----- NUMCOL - maximum number of generated columns
39     *----- NCORN - number of nods for each column (do not change)
40
41     *----- Argument declarations -----
42     *----- Global/External declarations -----
43     *----- Local declarations -----
44
45     ***** End listing of: CURRENTF.INC
46     C.....this routine calculates the "z" coordinate of a given yteste...
47     C.....on the surface of the ground.....
48     C.....the value of x is fixed and equals xfixed.....
49     INTEGER XFIXED,XX
50     REAL Y(MCOORD),Z(MSOIL,MCOORD,MCOORD)
51     XX=XFIXED
52     DO 10 J=1,NVERT
53         IF (Y(J).GT.YTESTE) THEN

```

^H

 PAGE 2
 04-12-91
 12:07:11

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

12      ALPHA=(Z(1,XX,J)-Z(1,XX,J-1))/(Y(J)-Y(J-1))
13      ZTESTE=Z(1,XX,J-1)+ALPHA*(YTESTE-Y(J-1))
14      RETURN
15      ELSE
16      IF (Y(J).EQ.YTESTE) THEN
17          ZTESTE=Z(1,XX,J)
18          ALPHA=(Z(1,XX,J+1)-Z(1,XX,J))/(Y(J+1)-Y(J))
19          RETURN
20      ELSE
21      END IF
22  END IF
23  10  CONTINUE
24  END

```

COORYZ Local Symbols

Name	Class	Type	Size	Offset
ALPHA	param			0006
ZTESTE	param			000a
YTESTE	param			000e
NVERT	param			0012
Z	param			0016
Y	param			001a
XFIXED	param			001e
J	local	INTEGER*4	4	0000
XX	local	INTEGER*4	4	0004

Symbolic Constant	Type	Value

MCOORD	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
COORYZ	FSUBRT	***	***	0000

Code size = 01c0 (448)

Data size = 0000 (0)

Bss size = 0008 (8)

No errors detected

^H

PAGE 1
04-12-91
12:07:43

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE CRITIC (NYPOL,NXPOL,XPOL,YPOL,ZPOL,NXSTAR,NXEND,NSURF,
2      1NSURFA,FS,NX,NY,X,Y,Z,NYSTAR,NYEND)
3      $include:'CURRENTF.INC'
4      ***** Begin listing of: CURRENTF.INC
5      *
6      *===== Prologue =====
7      *
8      * Purpose:
9      *   Include file with parameter declaration for the program PCSTL3D
10     *
11     * History:
12     *   Version          Programmer      Date          Description
13     *   -----
14     *   1.0              G.Moshkovich    ----          created
15     *
16     * IN args/commons          Units          Description
17     * -----
18     * OUT args/commons          Units          Description
19     * -----
20     * INOUT args/commons          Units          Description
21     * -----
22     * Processing:
23     *   none
24     *
25     * Special Requirements:
26     *   none
27     *
28     *----- Include files -----
29     *----- Constant declarations -----
30     integer MCOORD, MSOIL, NUMCOL, NCORN
31
32     parameter (MCOORD = 25, MSOIL = 5)
33     parameter (NUMCOL = 10, NCORN = 4)
34     PARAMETER (MAXSUR=2000)
35
36     *----- MCOORD - maximum number of points in each direction
37     *----- MSOIL - maximum number of soil layers
38     *----- NUMCOL - maximum number of generated columns
39     *----- NCORN - number of nuds for each column (do not change)
40
41     *----- Argument declarations -----
42     *----- Global/External declarations -----
43     *----- Local declarations -----
44
45     ***** End listing of: CURRENTF.INC
46     DIMENSION NXPOL1(NUMCOL,MCOORD), NYPOL1(NUMCOL), XPOL1(NUMCOL,
47     1MCOORD,MCOORD), YPOL1(NUMCOL,MCOORD), ZPOL1(NUMCOL,MCOORD,MCOORD)
48     2 NXSTA1(NUMCOL,MCOORD), NXEND1(NUMCOL,MCOORD), NXPOL(MCOORD),
49     3NXSTAR(MCOORD), NXEND(MCOORD), XPOL(MCOORD,MCOORD), YPOL(MCOORD),
50     4ZPOL(MCOORD,MCOORD), FS1(NUMCOL), X(MCOORD), Y(MCOORD), Z(MSOIL,
51     5MCOORD,MCOORD), NYSTA1(NUMCOL), NYEND1(NUMCOL)
52     CHARACTER NAMCRI*12
53     INTEGER*4 DIMEN1,DIMEN2

```

^H

 PAGE 2
 04-12-91
 12:07:43

```

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

12 PARAMETER (DIMEN1=NUMCOL*MCOORD,DIMEN2=NUMCOL*MCOORD*MCOORD)
13 DATA (NYPOL1(I),I=1,NUMCOL)/NUMCOL*1/
14 DATA ((NXPOL1(I,J),I=1,NUMCOL),J=1,MCOORD)/DIMEN1*1/
15 DATA (((XPOL1(I,J,K),I=1,NUMCOL),J=1,MCOORD),K=1,MCOORD)/DIMEN2*0
16 10/
17 DATA ((YPOL1(I,J),I=1,NUMCOL),J=1,MCOORD)/DIMEN1*0.0/
18 DATA (((ZPOL1(I,J,K),I=1,NUMCOL),J=1,MCOORD),K=1,MCOORD)/DIMEN2*0
19 10/
20 DATA ((NXSTA1(I,J),I=1,NUMCOL),J=1,MCOORD)/DIMEN1*1/
21 DATA ((NXEND1(I,J),I=1,NUMCOL),J=1,MCOORD)/DIMEN1*1/
22 DATA (FS1(I),I=1,NUMCOL)/NUMCOL*1000000.0/
23 DATA (NYSTA1(I),I=1,NUMCOL)/NUMCOL*1000000.0/
24 DATA (NYEND1(I),I=1,NUMCOL)/NUMCOL*1000000.0/
25 DO 80 I=1,NUMCOL
26 IF ((FS.LE.FS1(I)).AND.(FS.GT.0)) THEN
27 DO 40 J=NUMCOL,I+1,-1
28 FS1(J)=FS1(J-1)
29 NYPOL1(J)=NYPOL1(J-1)
30 NYSTA1(J)=NYSTA1(J-1)
31 NYEND1(J)=NYEND1(J-1)
32 DO 10 J1=1,NYPOL1(J-1)
33 NXPOL1(J,J1)=NXPOL1(J-1,J1)
34 YPOL1(J,J1)=YPOL1(J-1,J1)
35 NXSTA1(J,J1)=NXSTA1(J-1,J1)
36 NXEND1(J,J1)=NXEND1(J-1,J1)
37 10 CONTINUE
38 DO 30 J2=1,NYPOL1(J-1)
39 DO 20 J1=1,NXPOL1(J-1,J2)
40 XPOL1(J,J1,J2)=XPOL1(J-1,J1,J2)
41 ZPOL1(J,J1,J2)=ZPOL1(J-1,J1,J2)
42 20 CONTINUE
43 30 CONTINUE
44 40 CONTINUE
45 FS1(I)=FS
46 NYPOL1(I)=NYPOL
47 NYSTA1(I)=NYSTAR
48 NYEND1(I)=NYEND
49 DO 50 J1=1,NYPOL
50 NXPOL1(I,J1)=NXPOL(J1)
51 YPOL1(I,J1)=YPOL(J1)
52 NXSTA1(I,J1)=NXSTAR(J1)
53 NXEND1(I,J1)=NXEND(J1)
54 50 CONTINUE
55 DO 70 J2=1,NYPOL
56 DO 60 J1=1,NXPOL(J2)
57 XPOL1(I,J1,J2)=XPOL(J1,J2)
58 ZPOL1(I,J1,J2)=ZPOL(J1,J2)
59 60 CONTINUE
60 70 CONTINUE
61 C ---exit loop---
62 GO TO 90
63 END IF
64 80 CONTINUE
65 90 CONTINUE
66 C write(*,*) ' '
67 C do 3 i=1,NUMCOL
68 C write(*,*) ' fs(' ,i,') = ',fs1(i)
69 C 3 continue

```

^H

PAGE 3
04-12-91
12:07:43

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70      IF (NSURF.EQ.NSURFA) THEN
71 C      -----
72 C      PRINT DATA OF 10 MOST CRITICAL SURFACES
73 C      -----
74          WRITE (2,190)
75          DO 120 I=1,10
76              WRITE (2,200) I
77              WRITE (2,210)
78              DO 110 J=1,NYPOL1(I)
79                  DO 100 J1=1,NXPOL1(I,J)
80                      WRITE (2,220) XPOL1(I,J1,J),YPOL1(I,J),ZPOL1(I,J1
81 1                      J)
82 100      CONTINUE
83 110      CONTINUE
84          WRITE (2,230) FS1(I)
85 120      CONTINUE
86 C      -----
87 C      STORE DATA OF 2 MOST CRITICAL SURFACE TO PLOT
88 C      -----
89          WRITE (*,*) 'Please the most critical surface file name: '
90          READ (*,'(a)') NAMCRI
91          OPEN (3,FILE=NAMCRI)
92 C      open(4,file='plot10.dat')
93 C      open(5,file='aver.dat')
94 C.....for i= polygonal 1 (file for contour plot)
95          I=1
96          WRITE (3,*) 'PCSTBL3D v2.0 Critical Surface'
97          WRITE (3,*) NX,NY
98          DO 130 NNY=1,NYSTA1(I)-1
99              DO 130 NNX=1,NX
100              WRITE (3,240) X(NNX),Y(NNY),Z(1,NNX,NNY)
101 130      CONTINUE
102          DO 170 NNY=2,NYPOL1(I)-1
103              DO 140 NNX=1,NXSTA1(I,NNY)-1
104                  NYC=NYSTA1(I)+NNY-2
105                  WRITE (3,240) X(NNX),Y(NYC),Z(1,NNX,NYC)
106 140      CONTINUE
107          DO 150 J1=2,NXPOL1(I,NNY)-1
108              WRITE (3,240) XPOL1(I,J1,NNY),YPOL1(I,NNY),ZPOL1(I,J1
109 1          NNY)
110 150      CONTINUE
111          DO 160 NNX=NXEND1(I,NNY)+1,NX
112              NYC=NYSTA1(I)+NNY-2
113              WRITE (3,240) X(NNX),Y(NYC),Z(1,NNX,NYC)
114 160      CONTINUE
115 170      CONTINUE
116          DO 180 NNY=NYEND1(I)+1,NY
117              DO 180 NNX=1,NX
118                  WRITE (3,240) X(NNX),Y(NNY),Z(1,NNX,NNY)
119 180      CONTINUE
120      END IF
121      RETURN
122 C
123 190      FORMAT (///,7X,'TEN MOST CRITICAL SURFACES : ')
124 200      FORMAT (//,7X,'SURFACE # ',I2)
125 210      FORMAT (//,7X,'Coordinate  X',7X,'Coordinate  Y',7X,'Coordinate
126 1      Z',/)
127 220      FORMAT (9X,F8.3,13X,F8.3,13X,F8.3)

```


^H

PAGE 4
04-12-91
12:07:43

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

128 230 FORMAT (/ , 7X, 'Factor of Safety = ', F6.3)
129 240 FORMAT (1X, F8.3, 1X, F8.3, 1X, F8.3)
130      END

```

CRITIC Local Symbols

Name	Class	Type	Size	Offset
NYEND	param			0006
NYSTAR.	param			000a
Z	param			000e
Y	param			0012
X	param			0016
NY.	param			001a
NX.	param			001e
FS.	param			0022
NSURFA.	param			0026
NSURF	param			002a
NXEND	param			002e
NXSTAR.	param			0032
ZPOL.	param			0036
YPOL.	param			003a
XPOL.	param			003e
NXPOL	param			0042
NYPOL	param			0046
I	local	INTEGER*4	4	0002
J	local	INTEGER*4	4	0006
K	local	INTEGER*4	4	000a
NYC	local	INTEGER*4	4	000e
NNX	local	INTEGER*4	4	0012
NNY	local	INTEGER*4	4	0016
NAMCRI.	local	CHAR*12	12	001a
J1.	local	INTEGER*4	4	0026
J2.	local	INTEGER*4	4	002a
NYPOL1.	local	INTEGER*4	40	004a
NXPOL1.	local	INTEGER*4	1000	0072
XPOL1	local	REAL*4	25000	045a
YPOL1	local	REAL*4	1000	6602
ZPOL1	local	REAL*4	25000	69ea
NXSTA1.	local	INTEGER*4	1000	cb92
NXEND1.	local	INTEGER*4	1000	cf7a
FS1	local	REAL*4	40	d362
NYSTA1.	local	INTEGER*4	40	d38a
NYEND1.	local	INTEGER*4	40	d3b2

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000
DIMEN1.	INTEGER*4	250
DIMEN2.	INTEGER*4	6250

^H

PAGE 5
04-12-91
12:07:43

Microsoft FORTRAN Optimizing Compiler Version 5.00

Global Symbols

Name	Class	Type	Size	Offset
CRITIC.	FSUBRT	***	***	0000

Code size = 0b92 (2962)
Data size = d473 (54387)
Bss size = 002e (46)

No errors detected

^H

PAGE 1
04-12-91
12:07:55

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE FSCALC (COES,FI,A3,W3,U3,DIP,ALPHAYZ,FS,NYPOL,NCOLEF,
2      INCORIG)
3      $include:'CURRENTF.INC'
4      ***** Begin listing of: CURRENTF.INC
5
6      *
7      *===== Prologue =====
8      *
9      * Purpose:
10     *   Include file with parameter declaration for the program PCSTL3D
11     *
12     * History:
13     *   Version      Programmer      Date      Description
14     *   -----
15     *   1.0          G.Moshkovich      ----      created
16
17     * IN args/commons      Units      Description
18     * -----
19
20     * OUT args/commons      Units      Description
21     * -----
22
23     * INOUT args/commons      Units      Description
24     * -----
25
26     * Processing:
27     *   none
28
29     * Special Requirements:
30     *   none
31
32     *----- Include files -----
33     *----- Constant declarations -----
34     integer MCOORD, MSOIL, NUMCOL, NCORN
35
36     parameter (MCOORD = 25, MSOIL = 5)
37     parameter (NUMCOL = 10, NCORN = 4)
38     PARAMETER (MAXSUR=2000)
39
40     *----- MCOORD - maximum number of points in each direction
41     *----- MSOIL - maximum number of soil layers
42     *----- NUMCOL - maximum number of generated columns
43     *----- NCORN - number of nods for each column (do not change)
44
45     *----- Argument declarations -----
46     *----- Global/External declarations -----
47     *----- Local declarations -----
48
49     ***** End listing of: CURRENTF.INC
50
51     DIMENSION COES(MSOIL), FI(MSOIL), ALPHAYZ(MCOORD,-NUMCOL:NUMCOL),
52     1DIP(MCOORD,-NUMCOL:NUMCOL), A3(MCOORD,-NUMCOL:NUMCOL), W3(MCOORD,
53     2NUMCOL:NUMCOL), U3(MCOORD,-NUMCOL:NUMCOL), NCOLEF(MCOORD),
54     3NCORIG(MCOORD)
55     XNUMER=0.
56     DENOM=0.
57     DO 20 I=1,NYPOL-1
58         NNI=-NCOLEF(I)

```

^H

PAGE 2
04-12-91
12:07:55

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

12      NNF=NCORIG(I)
13      DO 10 J=NNI,NNF
14          IF (J.EQ.0) GO TO 10
15          EFFECT=(W3(I,J)*COS(DIP(I,J))-U3(I,J))
16          IF (EFFECT.LT.0.0) EFFECT=0.0
17          XNUMBER=XNUMBER+COES(1)*A3(I,J)+(EFFECT)*TAN(FI(1))
18          DENOM=DENOM+W3(I,J)*SIN(ALPHAYZ(I,J))
19      10  CONTINUE
20      20  CONTINUE
21      C ----- Factor of safety -----
22      FS1=XNUMBER/DENOM
23      C ----- Modified Factor of Safety -----
24      FS=FS1
25      C      if (fs1 .gt. 0) then
26      C          write(*,*) ' FS 3D = ', fs1
27      C      endif
28      ITER=1
29      30  XNUMBER=0.
30      DENOM=0.
31      DO 50 I=1,NYPOL-1
32          NNI=-NCOLEF(I)
33          NNF=NCORIG(I)
34          DO 40 J=NNI,NNF
35              IF (J.NE.0) THEN
36                  PES1=(W3(I,J)/COS(DIP(I,J)))-U3(I,J)*A3(I,J)
37      C          if (pes1 .lt. 0.0) pes1 = 0.0
38                  XITEM1=(PES1*TAN(FI(1))+COES(1)*A3(I,J))
39                  XITEM2=XITEM1*(COS(ALPHAYZ(I,J))+SIN(ALPHAYZ(I,J))*
40      1          TAN(DIP(I,J)))
41                  XITEM3=TAN(FI(1))*SIN(ALPHAYZ(I,J))/(FS*COS(DIP(I,J))
42                  XITEM4=(1.+XITEM3)
43                  XNUMBER=XNUMBER+XITEM2/XITEM4
44                  DENOM=DENOM+W3(I,J)*TAN(DIP(I,J))
45              END IF
46      40  CONTINUE
47      50  CONTINUE
48      FS2=XNUMBER/DENOM
49      IF (ABS(FS2-FS).LT.0.01) THEN
50      C          write(*,*) ' FS interactive = ',fs2
51          FS=FS2
52      END IF
53      IF ((ABS(FS2-FS).GE.0.01).AND.(ITER.LT.11)) THEN
54          FS=(FS2+FS)/2
55          ITER=ITER+1
56          GO TO 30
57      END IF
58      IF ((ITER.GE.11).AND.(ABS(FS2-FS).GE.0.01)) THEN
59      C          if (fs2.gt.0) write(*,*) 'FS did not converge..'
60      C          write(*,*) ' '
61      C          write(*,*) ' '
62          FS=FS1
63      END IF
64      RETURN
65      END

```

^H

PAGE 3
04-12-91
12:07:55

Microsoft FORTRAN Optimizing Compiler Version 5.00

FSCALC Local Symbols

Name	Class	Type	Size	Offset
NCORIG.	param			0006
NCOLEF.	param			000a
NYPOL	param			000e
FS.	param			0012
ALPHAYZ	param			0016
DIP	param			001a
U3.	param			001e
W3.	param			0022
A3.	param			0026
FI.	param			002a
COES.	param			002e
I	local	INTEGER*4	4	0000
FS1	local	REAL*4	4	0004
J	local	INTEGER*4	4	0008
FS2	local	REAL*4	4	000c
PES1.	local	REAL*4	4	0010
XNUMBER.	local	REAL*4	4	0014
NNF	local	INTEGER*4	4	0018
NNI	local	INTEGER*4	4	001c
EFFECT.	local	REAL*4	4	0020
DENOM	local	REAL*4	4	0024
ITER.	local	INTEGER*4	4	0028
XITEM1.	local	REAL*4	4	002c
XITEM2.	local	REAL*4	4	0030
XITEM3.	local	REAL*4	4	0034
XITEM4.	local	REAL*4	4	0038

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
FSCALC.	FSUBRT	***	***	0000

Code size = 0606 (1542)

Data size = 0010 (16)

Bss size = 003c (60)

No errors detected

^H

PAGE 1
04-12-91
12:08:21

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1 $nottruncate
2 $title:'PCSTBL3D v2.0'
3 $subtitle:'Input Data'
4 $large
5     SUBROUTINE INDATA (NX,NY,NSOIL,X,Y,Z,XAPRIN,YMIN1,DYMIN1,YMIN2,
6         1DYMIN2,XMIN1,DXMIN1,XMIN2,DXMIN2,ZMIN,DZMIN,GAMA,COES,FI,UCOND,RU
7         2UZ,DATOUT)
8 *
9 *===== Prologue =====
10 *
11 * Purpose:
12 *     Input data routine for the program PCSTBL3D
13 *
14 * History:
15 *     Version          Programmer      Date          Description
16 *     -----          -
17 *     1.0              J.E.Thomas      ----          created
18 *     2.0              G.Moshkovich    ----          code beautified,
19 *                                     ----          variables declared
20 *
21 * IN args/commons          Units          Description
22 * -----          -
23 *
24 * OUT args/commons          Units          Description
25 * -----          -
26 * nx
27 * ny
28 * nsoil
29 * x
30 * y
31 * z
32 * xaprin
33 * ymin1
34 * dymin1
35 * ymin2
36 * dymin2
37 * xmin1
38 * dxmin1
39 * xmin2
40 * dxmin2
41 * zmin
42 * dzmin
43 * gama
44 * coes
45 * fi
46 * ucond
47 * ru
48 * uz
49 * dataout
50 *
51 * INOUT args/commons          Units          Description
52 * -----          -
53 *
54 * Processing:
55 *     Open data file, check if the file is valid PCSTBL3D data file,
56 *     read geometry and soil data from the file, prints all data in the
57 *     output file defined by the user.
58 *

```

^H PCSTBL3D v2.0
Input Data

PAGE 2
04-12-91
12:08:21

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

59 * Special Requirements:
60 *   none
61 *
62 *----- Include files -----
63 $include: 'CURRENTF.INC'
**** Begin listing of: CURRENTF.INC
1  *
2  *----- Prologue -----
3  *
4  * Purpose:
5  *   Include file with parameter declaration for the program PCSTL3D
6  *
7  * History:
8  *   Version      Programmer      Date      Description
9  *   -----
10 *   1.0          G.Moshkovich      ----      created
11 *
12 * IN args/commons      Units      Description
13 * -----
14 *
15 * OUT args/commons      Units      Description
16 * -----
17 *
18 * INOUT args/commons      Units      Description
19 * -----
20 *
21 * Processing:
22 *   none
23 *
24 * Special Requirements:
25 *   none
26 *
27 *----- Include files -----
28 *----- Constant declarations -----
29     integer MCOORD, MSOIL, NUMCOL, NCORN
30
31     parameter (MCOORD = 25, MSOIL = 5)
32     parameter (NUMCOL = 10, NCORN = 4)
33     PARAMETER (MAXSUR=2000)
34
35 *----- MCOORD - maximum number of points in each direction
36 *----- MSOIL - maximum number of soil layers
37 *----- NUMCOL - maximum number of generated columns
38 *----- NCORN - number of nods for each column (do not change)
39
40 *----- Argument declarations -----
41 *----- Global/External declarations -----
42 *----- Local declarations -----
43
44
45
**** End listing of: CURRENTF.INC
64 *----- Constant declarations -----
65 *----- Argument declarations -----
66     REAL X(MCOORD), Y(MCOORD), XMIN1(MCOORD), DXMIN1(MCOORD),
67     1DXMIN2(MCOORD), XMIN2(MCOORD)
68     REAL Z(MSOIL, MCOORD, MCOORD), UZ(MSOIL, MCOORD, MCOORD)
69     REAL ZMIN(MCOORD, MCOORD), DZMIN(MCOORD, MCOORD)

```

^H PCSTBL3D v2.0
Input Data

PAGE 3
04-12-91
12:08:21

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70      REAL COES(MSOIL),GAMA(MSOIL),FI(MSOIL),RU(MSOIL)
71      INTEGER NX,NY,NSOIL
72      INTEGER UCOND(MSOIL)
73      CHARACTER*12 DATOUT
74      *----- Global/External declarations -----
75      *----- Local declarations -----
76      CHARACTER*12 DATIN
77      CHARACTER*20 TEST,LEGAL
78      INTEGER XAPRIN,I,J,K
79      *----- Code -----
80      *.....define input & output files.....
81      TEST='PCSTBL3D Version 2.0'
82      10  WRITE (*,50) ' Enter name of input file : '
83      READ (*,'(a)') DATIN
84      OPEN (1,FILE=DATIN)
85      READ (1,60) LEGAL
86      IF (LEGAL.NE.TEST) THEN
87          WRITE (*,50) ' Invalid data file, please reenter '
88          GO TO 10
89      END IF
90      WRITE (*,50) ' Enter name of output file : '
91      READ (*,'(a)') DATOUT
92      OPEN (2,FILE=DATOUT)
93      *.....start reading data.....
94      WRITE (2,70)
95      READ (1,*) NX,NY
96      WRITE (2,80) NX,NY
97      READ (1,*) NSOIL
98      WRITE (2,90) NSOIL
99      READ (1,*) (X(I),I=1,NX)
100     READ (1,*) (Y(J),J=1,NY)
101     WRITE (2,100) (I,X(I),I=1,NX)
102     WRITE (2,110) (J,Y(J),J=1,NY)
103     READ (1,*) ((Z(K,I,J),J=1,NY),I=1,NX),K=1,NSOIL)
104     DO 20 K=1,NSOIL
105         WRITE (2,120) K
106         WRITE (2,130) ((I,J,Z(K,I,J),J=1,NY),I=1,NX)
107     20  CONTINUE
108     WRITE (2,140)
109     READ (1,*) XAPRIN
110     WRITE (2,150) X(XAPRIN)
111     READ (1,*) YMIN1,DYMIN1,YMIN2,DYMIN2
112     WRITE (2,160) YMIN1,DYMIN1,YMIN2,DYMIN2
113     WRITE (2,170)
114     READ (1,*) (XMIN1(J),DXMIN1(J),XMIN2(J),DXMIN2(J),J=1,NY)
115     WRITE (2,180) (Y(J),XMIN1(J),DXMIN1(J),XMIN2(J),DXMIN2(J),J=1,NY)
116     WRITE (2,190)
117     READ (1,*) ((ZMIN(I,J),DZMIN(I,J),J=1,NY),I=1,NX)
118     WRITE (2,200) ((I,J,ZMIN(I,J),I,J,DZMIN(I,J),J=1,NY),I=1,NX)
119     WRITE (2,210)
120     READ (1,*) (GAMA(K),COES(K),FI(K),K=1,NSOIL)
121     WRITE (2,220) (K,GAMA(K),COES(K),FI(K),K=1,NSOIL)
122     *----- convert shear angle to radians
123     DO 30 K=1,NSOIL
124         FI(K)=FI(K)*0.01745329
125     30  CONTINUE
126     DO 40 K=1,NSOIL
127         READ (1,230) UCOND(K)

```


^H PCSTBL3D v2.0
Input Data

PAGE 4
04-12-91
12:08:21

```

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

128 IF (UCOND(K).EQ.1) THEN
129 READ (1,*) RU(K)
130 WRITE (2,240) K, RU(K)
131 ELSE
132 READ (1,*) ((UZ(K,I,J),J=1,NY),I=1,NX)
133 WRITE (2,250) ((I,J,UZ(K,I,J),J=1,NY),I=1,NX)
134 END IF
135 40 CONTINUE
136 RETURN
137 C
138 C
139 50 FORMAT (A)
140 60 FORMAT (20A)
141 70 FORMAT (1X, '*****'/'*
142 1 PC-STABL3D *'/'* 3-DIMENSIOANL SL
143 2PE STABILITY PROGRAM *'/'* FROM PURDUE UNIVERSITY
144 3 *'/'* VERSION 2.1 *'/'*****
145 4*****'/)
146 80 FORMAT (7X, 'Number of X coordinates : ', I3, '/', 7X, 'Number of Y coord
147 inates : ', I3)
148 90 FORMAT (7X, 'Number of soils : ', I3, '/')
149 100 FORMAT (//, 7X, 'X COORDINATES : ', //, (7X, 'coordinate X(', I3, ') = ',
150 1F7.2))
151 110 FORMAT (//, 7X, 'Y COORDINATES : ', //, (7X, 'coordinate Y(', I3, ') = ',
152 1F7.2,))
153 120 FORMAT (//, 7X, 'COORDINATES Z FOR SOIL # ', I2, '/')
154 130 FORMAT (7X, 'Z (', I2, ',', I2, ') = ', F7.2, 4X, 'Z (', I2, ',', I2, ') = ',
155 1F7.2, 4X, 'Z (', I2, ',', I2, ') = ', F7.2)
156 140 FORMAT (//, 7X, 'BOUNDARIES FOR GENERATION IN THE Y DIRECTION : ', //)
157 150 FORMAT (7X, 'Main Axis of Sliding was stablished at X = ', F7.2, '/')
158 160 FORMAT (7X, 'Minimum point for initiation of polygons at Y = ',
159 1F7.2, //, 7X, 'Interval for initiation of polygons is = ', F7.2, //, 7X,
160 2'Minimum point for termination of polygons at Y = ', F7.2, //, 7X,
161 3'interval for termination of polygons is = ', F7.2, //)
162 170 FORMAT (//, 7X, 'BOUNDARIES FOR GENERATION IN THE X DIRECTION : ', //)
163 180 FORMAT (//, 7X, 'Coordinate Y = ', F7.2, //, 7X, 'Minimum point for initi
164 tion of polygons at X = ', F7.2, //, 7X, 'Interval for initiation of
165 2polygons is = ', F7.2, //, 7X, 'Minimum point for termination of pol
166 3gonals at X = ', F7.2, //, 7X, 'Interval for termination of polygons
167 4is = ', F7.2, //)
168 190 FORMAT (//, 7X, 'BOUNDARIES FOR GENERATION IN THE Z DIRECTION : ', //)
169 200 FORMAT (7X, 'Zmin(', I2, ',', I2, ') = ', F7.2, 4X, 'Delta Z(', I2, ',', I2,
170 1) = ', F7.2)
171 210 FORMAT (//, 7X, 'SOIL PROPERTIES : ', //, 7X, 'Soil', 5X, 'Unit Weight', 5X
172 1'Cohesion', 5X, 'Shear Angle', //)
173 220 FORMAT (//, 8X, I2, 8X, F6.2, 8X, F7.1, 9X, F4.1)
174 230 FORMAT (I3)
175 240 FORMAT (//, 7X, 'Ru of soil #', I3, ' = ', F4.2)
176 250 FORMAT (//, 'PIEZOMETRIC LINE OF SOIL ', //, 7X, 'For x(', I2, ') and y
177 1', I2, '), U = ', F4.2))
178 END

```

INDATA Local Symbols

Name	Class	Type	Size	Offset
DATOUT.		param		0006

^H PCSTBL3D v2.0
Input Data

PAGE 5
04-12-91
12:08:21

Microsoft FORTRAN Optimizing Compiler Version 5.00

INDATA Local Symbols

Name	Class	Type	Size	Offset
UZ.	param			000a
RU.	param			000e
UCOND	param			0012
FI.	param			0016
COES.	param			001a
GAMA.	param			001e
DZMIN	param			0022
ZMIN.	param			0026
DXMIN2.	param			002a
XMIN2	param			002e
DXMIN1.	param			0032
XMIN1	param			0036
DYMIN2.	param			003a
YMIN2	param			003e
DYMIN1.	param			0042
YMIN1	param			0046
XAPRIN.	param			004a
Z	param			004e
Y	param			0052
X	param			0056
NSOIL	param			005a
NY.	param			005e
NX.	param			0062
TEST.	local	CHAR*20	20	0012
I	local	INTEGER*4	4	0026
J	local	INTEGER*4	4	002a
K	local	INTEGER*4	4	002e
LEGAL	local	CHAR*20	20	0032
DATIN	local	CHAR*12	12	0046

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
INDATA.	FSUBRT	***	***	0000

Code size = 1054 (4180)
Data size = 01ed (493)
Bss size = 0052 (82)

No errors detected

^H

PAGE 1
04-12-91
12:08:29

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      .SUBROUTINE PARAM1 (NYPOL,XCOL,YCOL,ZCOL,ZSOL,NCOLEF,NCORIG,
2      1ALPHAXZ,ALPHAYZ,DIP,TETA,A3,GAMA,RU,W3,U3,NSOIL)
3      $include:'CURRENTF.INC'
***** Begin listing of: CURRENTF.INC
1      *
2      *===== Prologue =====
3      *
4      * Purpose:
5      *   Include file with parameter declaration for the program PCSTL3D
6      *
7      * History:
8      *   Version          Programmer      Date          Description
9      *   -----          -
10     *   1.0              G.Moshkovich    ----          created
11     *
12     * IN args/commons          Units          Description
13     * -----
14     *
15     * OUT args/commons          Units          Description
16     * -----
17     *
18     * INOUT args/commons          Units          Description
19     * -----
20     *
21     * Processing:
22     *   none
23     *
24     * Special Requirements:
25     *   none
26     *
27     *----- Include files -----
28     *----- Constant declarations -----
29     integer MCOORD, MSOIL, NUMCOL, NCORN
30
31     parameter (MCOORD = 25, MSOIL = 5)
32     parameter (NUMCOL = 10, NCORN = 4)
33     PARAMETER (MAXSUR=2000)
34
35     *----- MCOORD - maximum number of points in each direction
36     *----- MSOIL - maximum number of soil layers
37     *----- NUMCOL - maximum number of generated columns
38     *----- NCORN - number of nods for each column (do not change)
39
40     *----- Argument declarations -----
41     *----- Global/External declarations -----
42     *----- Local declarations -----
43
44
45
***** End listing of: CURRENTF.INC
4      DIMENSION XCOL(MCOORD,-NUMCOL:NUMCOL,NCORN), YCOL(MCOORD,-
5      1NUMCOL:NUMCOL,NCORN), ZCOL(MCOORD,-NUMCOL:NUMCOL,NCORN),
6      2NCOLEF(MCOORD), NCORIG(MCOORD), ALPHAXZ(MCOORD,-NUMCOL:NUMCOL),
7      3ALPHAYZ(MCOORD,-NUMCOL:NUMCOL), DIP(MCOORD,-NUMCOL:NUMCOL),
8      4TETA(MCOORD,-NUMCOL:NUMCOL), A3(MCOORD,-NUMCOL:NUMCOL),
9      5ZSOL(MSOIL,MCOORD,-NUMCOL:NUMCOL,NCORN), W3(MCOORD,-NUMCOL:NUMCOL
10     6, U3(MCOORD,-NUMCOL:NUMCOL), GAMA(MSOIL), RU(MSOIL)
11 C      -----

```

^H

PAGE 2
04-12-91
12:08:29

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

12 C      Calculation of column parameters
13 C      -----
14      DO 30 I=1,NYPOL-1
15          NNI=-NCOLEF(I)
16          NNF=NCORIG(I)
17          DO 20 J=NNI,NNF
18              IF (J.EQ.0) GO TO 20
19              DELTAZ1=(ZCOL(I,J,1)+ZCOL(I,J,2)-(ZCOL(I,J,3)+ZCOL(I,J,4)
20                  )/2.
21              DELTAX1=XCOL(I,J,1)-XCOL(I,J,4)
22 C      ---- alphaxz(i,j) ----
23              ALPHAXZ(I,J)=ATAN(DELTAZ1/DELTAX1)
24              DELTAZ2=(ZCOL(I,J,2)+ZCOL(I,J,3)-(ZCOL(I,J,1)+ZCOL(I,J,4)
25                  )/2.
26              DELTAY2=(YCOL(I,J,2)+YCOL(I,J,3)-(YCOL(I,J,1)+YCOL(I,J,4)
27                  )/2.
28 C      ---- alphayz(i,j) ----
29              ALPHAYZ(I,J)=ATAN(DELTAZ2/DELTAY2)
30 C      ---- DIP(i,j) ----
31              DIP(I,J)=ACOS(1./SQRT(1+(TAN(ALPHAYZ(I,J)))**2+
32                  (TAN(ALPHAXZ(I,J)))**2))
33 C      ---- teta(i,j) and Area A3(i,j) ----
34              TETA(I,J)=SQRT(1.-((SIN(ALPHAXZ(I,J)))**2)*(SIN(ALPHAYZ(I,
35                  J)))**2))
36              A3(I,J)=(DELTAX1*DELTAY2*TETA(I,J))/(COS(ALPHAXZ(I,J))*
37                  COS(ALPHAYZ(I,J)))
38 C      ---- W3(i,j) = weight of the column ----
39              W3(I,J)=0.0
40              ZZCOL=ZCOL(I,J,1)+ZCOL(I,J,2)+ZCOL(I,J,3)+ZCOL(I,J,4)
41              ZZCOL=ZZCOL/4.
42              IUCODE=0
43              DO 10 K=NSOIL,1,-1
44                  ZZ1=ZSOL(K,I,J,1)+ZSOL(K,I,J,2)+ZSOL(K,I,J,3)+ZSOL(K,
45                      I,J,4)
46                  ZZ1=ZZ1/4.
47                  IF (ZZ1.GT.ZZCOL) THEN
48                      W3(I,J)=W3(I,J)+(ZZ1-ZZCOL)*GAMA(K)
49                      ZZCOL=ZZ1
50                      IF (IUCODE.EQ.0) THEN
51                          IUCODE=K
52                      END IF
53                  END IF
54      10      CONTINUE
55              U3(I,J)=W3(I,J)*RU(IUCODE)
56              W3(I,J)=W3(I,J)*DELTAX1*DELTAY2
57      20      CONTINUE
58      30      CONTINUE
59      RETURN
60      END

```

PARAM1 Local Symbols

Name	Class	Type	Size	Offset
NSOIL	param			0006
U3.	param			000a
W3.	param			000e

^H

PAGE 3
04-12-91
12:08:29

Microsoft FORTRAN Optimizing Compiler Version 5.00

PARAM1 Local Symbols

Name	Class	Type	Size	Offset
RU.	param			0012
GAMA.	param			0016
A3.	param			001a
TETA.	param			001e
DIP	param			0022
ALPHAYZ	param			0026
ALPHAXZ	param			002a
NCORIG.	param			002e
NCOLEF.	param			0032
ZSOL.	param			0036
ZCOL.	param			003a
YCOL.	param			003e
XCOL.	param			0042
NYPOL	param			0046
I	local	INTEGER*4	4	0000
J	local	INTEGER*4	4	0004
K	local	INTEGER*4	4	0008
ZZCOL	local	REAL*4	4	000c
NNF	local	INTEGER*4	4	0010
ZZ1	local	REAL*4	4	0014
NNI	local	INTEGER*4	4	0018
DELTAX1	local	REAL*4	4	001c
DELTAY2	local	REAL*4	4	0020
DELTAZ1	local	REAL*4	4	0024
DELTAZ2	local	REAL*4	4	0028
IUCODE.	local	INTEGER*4	4	002c

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
PARAM1.	FSUBRT	***	***	0000

Code size = 04af (1199)

Data size = 000c (12)

Bss size = 0030 (48)

No errors detected

^H

PAGE 1
04-12-91
12:08:37

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE POLYGO (NPPOLY,XVPOL,YVERTICE,DELTA,COD,DIFF,K1,RANDC,
2      1RANDL,RANDR,XROW,Y,Z,NY,YVPOL)
3      INTEGER XROW
4      $include:'CURRENTF.INC'
5      ***** Begin listing of: CURRENTF.INC
6
7      *
8      *===== Prologue =====
9      *
10     * Purpose:
11     *   Include file with parameter declaration for the program PCSTL3D
12     *
13     * History:
14     *   Version      Programmer      Date      Description
15     *   -----
16     *   1.0          G.Moshkovich      created
17     *
18     * IN args/commons      Units      Description
19     * -----
20     *
21     * OUT args/commons      Units      Description
22     * -----
23     *
24     * INOUT args/commons      Units      Description
25     * -----
26     *
27     * Processing:
28     *   none
29     *
30     * Special Requirements:
31     *   none
32     *
33     *----- Include files -----
34     *----- Constant declarations -----
35     integer MCOORD, MSOIL, NUMCOL, NCORN
36
37     parameter (MCOORD = 25, MSOIL = 5)
38     parameter (NUMCOL = 10, NCORN = 4)
39     PARAMETER (MAXSUR=2000)
40
41     *----- MCOORD - maximum number of points in each direction
42     *----- MSOIL - maximum number of soil layers
43     *----- NUMCOL - maximum number of generated columns
44     *----- NCORN - number of nods for each column (do not change)
45
46     *----- Argument declarations -----
47     *----- Global/External declarations -----
48     *----- Local declarations -----
49
50     ***** End listing of: CURRENTF.INC
51     REAL K1,N30,N45,N85,N89,T(30),YVERTICE(28),DELTA(MCOORD),
52     1XVPOL(MCOORD),YVPOL(MCOORD),Y(MCOORD),Z(MSOIL,MCOORD,MCOORD)
53     CHARACTER COD
54     C.....this routine generates random surfaces.....
55     C.....define values in radians.....
56     N30=30*3.141593/180
57     N45=45*3.141593/180

```

^H

 PAGE 2
 04-12-91
 12:08:37

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

12      N85=85*3.141593/180
13      N89=89*3.141593/180
14      T(1)=0
15      NPPOL=NPPOLY+1
16      DO 30 I=2,NPPOLY
17          NCONT=0
18          J=I+DIFF
19          IF (COD.EQ.'c') THEN
20              CALL RNDC (RANDC)
21              RAND=RANDC
22          END IF
23          IF (COD.EQ.'l') THEN
24              CALL RNDL (RANDL)
25              RAND=RANDL
26          END IF
27          IF (COD.EQ.'r') THEN
28              CALL RNDR (RANDR)
29              RAND=RANDR
30          END IF
31      C.....If ny=0 it means that we are calculating the body of the.....
32      C.....polygonal.....
33          IF ((I.EQ.2).AND.((COD.EQ.'c').OR.(NY.EQ.0))) THEN
34              IF (COD.NE.'c') THEN
35                  T(2)=N30*RAND**2
36                  YVPOL(2)=YVPOL(1)+(XVPOL(2)-XVPOL(1))*TAN(T(2))
37              ELSE
38                  T(2)=-N45+(0.95*ATAN(K1)+N45)*RAND**2
39                  YVPOL(2)=YVPOL(1)+(XVPOL(2)-XVPOL(1))*TAN(T(2))
40              END IF
41          ELSE
42              IF (COD.EQ.'c') THEN
43                  CALL RNDC (RANDC)
44                  RAND=RANDC
45              END IF
46              IF (COD.EQ.'l') THEN
47                  CALL RNDL (RANDL)
48                  RAND=RANDL
49              END IF
50              IF (COD.EQ.'r') THEN
51                  CALL RNDR (RANDR)
52                  RAND=RANDR
53              END IF
54      10      YVPOL(I)=YVERTICE(J)+RAND*DELTA(J)
55              T(I)=ATAN((YVPOL(I)-YVPOL(I-1))/(XVPOL(I)-XVPOL(I-1)))
56              IF (I.EQ.2) GO TO 20
57              IF (T(I).GT.N85) THEN
58                  IF (COD.EQ.'c') THEN
59                      CALL RNDC (RANDC)
60                      RAND=RANDC
61                  END IF
62                  IF (COD.EQ.'l') THEN
63                      CALL RNDL (RANDL)
64                      RAND=RANDL
65                  END IF
66                  IF (COD.EQ.'r') THEN
67                      CALL RNDR (RANDR)
68                      RAND=RANDR
69                  END IF

```

^H

PAGE 3
04-12-91
12:08:37

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70          GO TO 10
71      END IF
72      XTESTE=XVPOL(1)+2*(XVPOL(NPPOL)-XVPOL(1))/3
73      IF ((XVPOL(I).GT.XTESTE).AND.(T(I).LT.T(I-1))) THEN
74          TTESTE=ATAN((YVERTICE(J)+DELTA(J)-YVPOL(I-1))/
75              1      (XVPOL(I)-XVPOL(I-1)))
76          IF ((TTESTE.GE.T(I-1)).AND.(NCONT.LE.5)) THEN
77              NCONT=NCONT+1
78              IF (COD.EQ.'c') THEN
79                  CALL RND C (RANDC)
80                  RAND=RANDC
81              END IF
82              IF (COD.EQ.'1') THEN
83                  CALL RND L (RANDL)
84                  RAND=RANDL
85              END IF
86              IF (COD.EQ.'r') THEN
87                  CALL RND R (RANDR)
88                  RAND=RANDR
89              END IF
90              GO TO 10
91      ELSE
92          YVPOL(I)=YVERTICE(J)+DELTA(J)
93          T(I)=ATAN((YVPOL(I)-YVPOL(I-1))/(XVPOL(I)-XVPOL(I
94      1      1)))
95      END IF
96  ELSE
97  END IF
98  IF (T(I).LE.T(I-1)) THEN
99      IF (COD.NE.'c') THEN
100          T(I)=T(I-1)
101      ELSE
102          IF ((T(I)-T(I-1)).LT.(-N45*RAND**RAND)) THEN
103              V=T(I-1)-N45*RAND**RAND
104              IF (V.GT.(-N45)) THEN
105                  T(I)=V
106              ELSE
107                  T(I)=-N45
108              END IF
109          END IF
110      END IF
111  ELSE
112      IF ((T(I)-T(I-1)).GE.(N45*RAND**RAND)) THEN
113          IF ((T(I-1)+N45*RAND**RAND).GT.N85) THEN
114              T(I)=N85
115          ELSE
116              T(I)=T(I-1)+N45*RAND**RAND
117          END IF
118      END IF
119  END IF
120      YVPOL(I)=YVPOL(I-1)+(XVPOL(I)-XVPOL(I-1))*TAN(T(I))
121  END IF
122  C.....now , check if yvpol is within limits.....
123      20      IF (YVPOL(I).LT.YVERTICE(J)) YVPOL(I)=YVERTICE(J)
124          IF (YVPOL(I).GT.(YVERTICE(J)+DELTA(J))) YVPOL(I)=YVERTICE(J)+
125      1      DELTA(J)
126          T(I)=ATAN((YVPOL(I)-YVPOL(I-1))/(XVPOL(I)-XVPOL(I-1)))
127  C.....final of "DO" procedure.....

```


^H

 PAGE 4
 04-12-91
 12:08:37

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

128 30 CONTINUE
129 IF (COD.NE.'c') GO TO 60
130 40 T(NPPOL)=ATAN((YVPOL(NPPOL)-YVPOL(NPPOLY))/(XVPOL(NPPOL)-
131 1XVPOL(NPPOLY)))
132 IF (T(NPPOL).GT.N89) THEN
133 XVPOL(NPPOL)=XVPOL(NPPOL)+1
134 XTESTE=XVPOL(NPPOL)
135 CALL COORYZ (XROW,Y,Z,NY,XTESTE,YTESTE,ALPHA)
136 YVPOL(NPPOL)=YTESTE
137 GO TO 40
138 ELSE
139 END IF
140 50 IF (T(NPPOL).LT.T(NPPOLY)) THEN
141 XVPOL(NPPOL)=XVPOL(NPPOL)-1
142 XTESTE=XVPOL(NPPOL)
143 CALL COORYZ (XROW,Y,Z,NY,XTESTE,YTESTE,ALPHA)
144 YVPOL(NPPOL)=YTESTE
145 T(NPPOL)=ATAN((YVPOL(NPPOL)-YVPOL(NPPOLY))/(XVPOL(NPPOL)-
146 1 XVPOL(NPPOLY)))
147 GO TO 50
148 ELSE
149 END IF
150 60 RETURN
151 END

```

POLYGO Local Symbols

Name	Class	Type	Size	Offset
YVPOL	param			0006
NY	param			000a
Z	param			000e
Y	param			0012
XROW	param			0016
RANDR	param			001a
RANDL	param			001e
RANDC	param			0022
K1	param			0026
DIFF	param			002a
COD	param			002e
DELTA	param			0032
YVERTICE	param			0036
XVPOL	param			003a
NPPOLY	param			003e
NCONT	local	INTEGER*4	4	0000
I	local	INTEGER*4	4	0004
NPPOL	local	INTEGER*4	4	0008
J	local	INTEGER*4	4	000c
T	local	REAL*4	120	0010
V	local	REAL*4	4	0088
TTESTE	local	REAL*4	4	008c
XTESTE	local	REAL*4	4	0090
YTESTE	local	REAL*4	4	0094
RAND	local	REAL*4	4	0098
ALPHA	local	REAL*4	4	009c
N30	local	REAL*4	4	00a0
N45	local	REAL*4	4	00a4

^H

PAGE 5
04-12-91
12:08:37

Microsoft FORTRAN Optimizing Compiler Version 5.00

POLYGO Local Symbols

Name	Class	Type	Size	Offset
N85	local	REAL*4	4	00a8
N89	local	REAL*4	4	00ac

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

Global Symbols

Name	Class	Type	Size	Offset
COORYZ.	extern	***	***	***
POLYGO.	FSUBRT	***	***	0000
RNDC.	extern	***	***	***
RNDL.	extern	***	***	***
RNDR.	extern	***	***	***

Code size = 0931 (2353)
Data size = 0030 (48)
Bss size = 00b0 (176)

No errors detected

^H

PAGE 1
04-12-91
12:08:43

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE RNDC (X)
2      *
3      *===== Prologue =====
4      *
5      * Purpose:
6      *   Generate random number numbers for the axis
7      *
8      * History:
9      *   Version      Programmer      Date      Description
10     *   -----
11     *   1.0          J.E.Thomas      ----      created
12     *   2.0          G.Moshkovich    ----      code beautified,
13     *                                     variables declared
14     *
15     * IN args/commons      Units      Description
16     * -----
17     *
18     * OUT args/commons     Units      Description
19     * -----
20     *
21     * INOUT args/commons   Units      Description
22     * -----
23     * x                    N/A        Generated random number
24     *
25     * Processing:
26     *   Convert seed and given numbers to the pseudo-random numbers
27     * with integer and fractional part processing.
28     *
29     * Special Requirements:
30     *   none
31     *
32     *----- Include files -----
33     *----- Constant declarations -----
34     *----- Argument declarations -----
35     REAL X
36     *----- Global/External declarations -----
37     *----- Local declarations -----
38     REAL RM
39     INTEGER K,J,M,IX,IRAND
40     DATA K,J,M,RM/5701,3612,566927,566927.0/
41     *----- Code -----
42     IX=INT(X*RM)
43     IRAND=MOD(J*IX+K,M)
44     X=(REAL(IRAND)+0.5)/RM
45     RETURN
46     END

```

RNDC Local Symbols

Name	Class	Type	Size	Offset
X	param			0006
K	local	INTEGER*4	4	0000
IX.	local	INTEGER*4	4	0000
J	local	INTEGER*4	4	0004
IRAND	local	INTEGER*4	4	0004
M	local	INTEGER*4	4	0008

^H

PAGE 2
04-12-91
12:08:43

Microsoft FORTRAN Optimizing Compiler Version 5.00

RNDC Local Symbols

Name	Class	Type	Size	Offset
RM.	local	REAL*4	4	000c

Global Symbols

Name	Class	Type	Size	Offset
RNDC.	FSUBRT	***	***	0000

Code size = 0066 (102)

Data size = 0014 (20)

Bss size = 0008 (8)

No errors detected

^H

 PAGE 1
 04-12-91
 12:08:48

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1 $title:'PCSTBL3D v2.0'
2 $subtitle:'Random Left Side Numbers'
3 SUBROUTINE RNDL (X)
4 *
5 *===== Prologue =====
6 *
7 * Purpose:
8 *   Generate random number numbers for the left side of the 3D surface
9 *
10 * History:
11 *   Version      Programmer      Date      Description
12 *   -----
13 *   1.0          J.E.Thomas
14 *   2.0          G.Moshkovich     created
15 *                                     code beautified,
16 *                                     variables declared
17 * IN args/commons      Units      Description
18 * -----
19 *
20 * OUT args/commons      Units      Description
21 * -----
22 *
23 * INOUT args/commons    Units      Description
24 * -----
25 * x                     N/A       Generated random number
26 *
27 * Processing:
28 *   Convert seed and given numbers to the pseudo-random numbers
29 * with integer and fractional part processing.
30 *
31 * Special Requirements:
32 *   none
33 *
34 *----- Include files -----
35 *----- Constant declarations -----
36 *----- Argument declarations -----
37 REAL X
38 *----- Global/External declarations -----
39 *----- Local declarations -----
40 REAL RM
41 INTEGER K,J,M,IX,IRAND
42 DATA K,J,M,RM/5701,3612,566927,566927.0/
43 *----- Code -----
44 IX=INT(X*RM)
45 IRAND=MOD(J*IX+K,M)
46 X=(REAL(IRAND)+0.5)/RM
47 RETURN
48 END

```

RNDL Local Symbols

Name	Class	Type	Size	Offset
X	param			0006
K	local	INTEGER*4	4	0000
IX.	local	INTEGER*4	4	0000
J	local	INTEGER*4	4	0004

^H PCSTBL3D v2.0
Random Left Side Numbers

PAGE 2
04-12-91
12:08:48

Microsoft FORTRAN Optimizing Compiler Version 5.00

RNDL Local Symbols

Name	Class	Type	Size	Offset
IRAND	local	INTEGER*4	4	0004
M	local	INTEGER*4	4	0008
RM.	local	REAL*4	4	000c

Global Symbols

Name	Class	Type	Size	Offset
RNDL.	FSUBRT	***	***	0000

Code size = 0066 (102)
Data size = 0014 (20)
Bss size = 0008 (8)

No errors detected

^H

PAGE 1
04-12-91
12:08:53

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE RNDR (X)
2      *
3      *===== Prologue =====
4      *
5      * Purpose:
6      *   Generate random number numbers for the right side of the 3D surfac
7      *
8      * History:
9      *   Version      Programmer      Date      Description
10     *   -----      -
11     *   1.0          J.E.Thomas      created
12     *   2.0          G.Moshkovich    code beautified,
13     *                                     variables declared
14     *
15     * IN args/commons      Units      Description
16     * -----      -
17     *
18     * OUT args/commons      Units      Description
19     * -----      -
20     *
21     * INOUT args/commons      Units      Description
22     * -----      -
23     * x                      N/A      Generated random number
24     *
25     * Processing:
26     *   Convert seed and given numbers to the pseudo-random numbers
27     * with integer and fractional part processing.
28     *
29     * Special Requirements:
30     *   none
31     *
32     *----- Include files -----
33     *----- Constant declarations -----
34     *----- Argument declarations -----
35     REAL X
36     *----- Global/External declarations -----
37     *----- Local declarations -----
38     REAL RM
39     INTEGER K,J,M,IX,IRAND
40     DATA K,J,M,RM/5701,3612,566927,566927.0/
41     *----- Code -----
42     IX=INT(X*RM)
43     IRAND=MOD(J*IX+K,M)
44     X=(REAL(IRAND)+0.5)/RM
45     RETURN
46     END

```

RNDR Local Symbols

Name	Class	Type	Size	Offset
X	param			0006
K	local	INTEGER*4	4	0000
IX.	local	INTEGER*4	4	0000
J	local	INTEGER*4	4	0004
IRAND	local	INTEGER*4	4	0004
M	local	INTEGER*4	4	0008

^H

PAGE 2
04-12-91
12:08:53

Microsoft FORTRAN Optimizing Compiler Version 5.00

RNDR Local Symbols

Name	Class	Type	Size	Offset
RM.	local	REAL*4	4	000c

Global Symbols

Name	Class	Type	Size	Offset
RNDR.	FSUBRT	***	***	0000

Code size = 0066 (102)

Data size = 0014 (20)

Bss size = 0008 (8)

No errors detected

^H

PAGE 1
04-12-91
12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1 C---Subroutine Wibul calculates and prints out minimum
2 C---statistical value of the factor of safety
3     SUBROUTINE WEIBUL (NFS,FSCUM)
4     IMPLICIT REAL (A-Z)
5     INTEGER NFS,I,II
6     DIMENSION FSCUM(*)
7     $large: e
8     DIMENSION E(4000)
9     C      ----- Weibulls Calculations -----
10    DO 10 I=1,NFS
11        IF (FSCUM(I).LT.399) THEN
12            MAXFS=MAX(MAXFS,FSCUM(I))
13        END IF
14    10 CONTINUE
15    DO 30 I=1,NFS
16        DO 20 II=1,INT(MAXFS+1)
17            IF (FSCUM(I).LE.II/10.) THEN
18                E(II)=E(II)+1.
19                GO TO 30
20            END IF
21    20 CONTINUE
22    30 CONTINUE
23        E2=0
24        E3=0
25        DO 40 I=1,INT(MAXFS+1)
26            E2=E2+E(I)*I/10.
27            E3=E3+E(I)
28    40 CONTINUE
29        E6=E2/E3
30        E2=0
31        E4=0
32        DO 50 I=1,INT(MAXFS+1)
33            E2=E2+E(I)*(I/10.-E6)**2
34            E4=E4+E(I)*(I/10.-E6)**3
35    50 CONTINUE
36        E7=SQRT(E2/E3)
37        E8=E4/E3
38        E9=E8/E7**3
39    C      for b1= e9 get the value of 1/B
40        CALL FUNCT1 (B,E9)
41        B1=1+B
42        B2=1+2*B
43        N=B1
44        G1=GAMMA(N)
45        N=B2
46        G2=GAMMA(N)
47        B4=E6-E7*G1/SQRT(G2-G1*G1)
48        B3=(E6-B4)/G1+B4
49        B5=1/B
50    C---F = ((I1 - B4) / (B3 - B4)) ^ (B5 - 1) * B5 / (B3 - B4) *c
51    C---EXP(-((I1 - B4) / (B3 - B4)) ^ B5)
52        PRINT 60, 'MINIMUM STATISTICAL FACTOR OF SAFETY IS ',B4
53        IF (B4.LE.0.) THEN
54            PRINT *
55            PRINT *
56            PRINT *, '          ***ATTENTION***'
57            PRINT *, 'MINIMUM STATISTICAL FACTOR OF SAFETY IS NEGATIVE'
58            PRINT *, 'REVIEW YOUR SURFACE GENERATION LIMITS OR INCREASE'

```

^H

PAGE 2
04-12-91
12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

59      PRINT *, '          NUMBER OF GENERATED SURFACES'
60      PRINT *
61      ELSE IF (B4.LE.0.1) THEN
62          PRINT *
63          PRINT *
64          PRINT *, '          ***ATTENTION***'
65          PRINT *, 'MINIMUM STATISTICAL FACTOR OF SAFETY IS VERY SMALL'
66          PRINT *, 'REVIEW YOUR SURFACE GENERATION LIMITS OR INCREASE'
67          PRINT *, '          NUMBER OF GENERATED SURFACES'
68          PRINT *
69      END IF
70      RETURN
71  C
72  60  FORMAT (//,1X,A,F5.2,/)
73      END

```

WEIBUL Local Symbols

Name	Class	Type	Size	Offset
FSCUM	param			0006
NFS	param			000a
E	local	REAL*4	16000	0000
B	local	REAL*4	4	0002
I	local	INTEGER*4	4	0006
N	local	REAL*4	4	000a
II.	local	INTEGER*4	4	000e
B1.	local	REAL*4	4	0012
B2.	local	REAL*4	4	0016
B3.	local	REAL*4	4	001a
B4.	local	REAL*4	4	001e
B5.	local	REAL*4	4	0022
E2.	local	REAL*4	4	0026
G1.	local	REAL*4	4	002a
E3.	local	REAL*4	4	002e
G2.	local	REAL*4	4	0032
E4.	local	REAL*4	4	0036
E6.	local	REAL*4	4	003a
E7.	local	REAL*4	4	003e
E8.	local	REAL*4	4	0042
E9.	local	REAL*4	4	0046
MAXFS	local	REAL*4	4	004a

```

74      REAL FUNCTION GAMMA(X)
75  CD    DOUBLE PRECISION FUNCTION DGAMMA(X)

```

```

76  C-----
77  C
78  C This routine calculates the GAMMA function for a real argument X.
79  C Computation is based on an algorithm outlined in reference 1.
80  C The program uses rational functions that approximate the GAMMA
81  C function to at least 20 significant decimal digits. Coefficients
82  C for the approximation over the interval (1,2) are unpublished.
83  C Those for the approximation for X .GE. 12 are from reference 2.
84  C The accuracy achieved depends on the arithmetic system, the
85  C compiler, the intrinsic functions, and proper selection of the
86  C machine-dependent constants.
87  C

```

^H

PAGE 3
04-12-91
12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

88 C
89 C*****
90 C*****
91 C
92 C Explanation of machine-dependent constants
93 C
94 C beta    - radix for the floating-point representation
95 C maxexp  - the smallest positive power of beta that overflows
96 C XBIG    - the largest argument for which GAMMA(X) is representable
97 C          in the machine, i.e., the solution to the equation
98 C          GAMMA(XBIG) = beta**maxexp
99 C XINF    - the largest machine representable floating-point number;
100 C          approximately beta**maxexp
101 C EPS     - the smallest positive floating-point number such that
102 C          1.0+EPS .GT. 1.0
103 C XMININ  - the smallest positive floating-point number such that
104 C          1/XMININ is machine representable
105 C
106 C      Approximate values for some important machines are:
107 C
108 C          beta          maxexp          XBIG
109 C
110 C CRAY-1          (S.P.)          2          8191          966.961
111 C Cyber 180/855  (S.P.)          2          1070          177.803
112 C   under NOS    (S.P.)          2          128          35.040
113 C IEEE (IBM/XT,  (S.P.)          2          1024          171.624
114 C   SUN, etc.)   (D.P.)          16          63          57.574
115 C IEEE (IBM/XT,  (D.P.)          2          127          34.844
116 C   SUN, etc.)   (D.P.)          2          1023          171.489
117 C IBM 3033       (D.P.)          2
118 C VAX D-Format   (D.P.)          2
119 C VAX G-Format   (D.P.)          2
120 C
121 C          XINF          EPS          XMININ
122 C
123 C CRAY-1          (S.P.)  5.45E+2465  7.11E-15  1.84E-2466
124 C Cyber 180/855  (S.P.)  1.26E+322  3.55E-15  3.14E-294
125 C   under NOS    (S.P.)  3.40E+38   1.19E-7   1.18E-38
126 C IEEE (IBM/XT,  (S.P.)  1.79D+308  2.22D-16  2.23D-308
127 C   SUN, etc.)   (D.P.)  7.23D+75   2.22D-16  1.39D-76
128 C IEEE (IBM/XT,  (D.P.)  1.70D+38   1.39D-17  5.88D-39
129 C   SUN, etc.)   (D.P.)  8.98D+307  1.11D-16  1.12D-308
130 C IBM 3033       (D.P.)
131 C VAX D-Format   (D.P.)
132 C VAX G-Format   (D.P.)
133 C
134 C*****
135 C*****
136 C
137 C Error returns
138 C
139 C The program returns the value XINF for singularities or
140 C when overflow would occur. The computation is believed
141 C to be free of underflow and overflow.
142 C
143 C
144 C Intrinsic functions required are:
145 C

```

^H

PAGE 4
04-12-91
12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

146 C      INT, DBLE, EXP, LOG, REAL, SIN
147 C
148 C
149 C References: "An Overview of Software Development for Special
150 C             Functions", W. J. Cody, Lecture Notes in Mathematics,
151 C             506, Numerical Analysis Dundee, 1975, G. A. Watson
152 C             (ed.), Springer Verlag, Berlin, 1976.
153 C
154 C             Computer Approximations, Hart, Et. Al., Wiley and
155 C             sons, New York, 1968.
156 C
157 C Latest modification: October 12, 1989
158 C
159 C Authors: W. J. Cody and L. Stoltz
160 C             Applied Mathematics Division
161 C             Argonne National Laboratory
162 C             Argonne, IL 60439
163 C
164 C-----
165 C      INTEGER I,N
166 C      LOGICAL PARITY
167 C      REAL
168 CD     DOUBLE PRECISION
169 C      1      C, CONV, EPS, FACT, HALF, ONE, P, PI, Q, RES, SQRTPI, SUM, TWELVE,
170 C      2      TWO, X, XBIG, XDEN, XINF, XMININ, XNUM, Y, Y1, YSQ, Z, ZERO
171 C      DIMENSION C(7), P(8), Q(8)
172 C-----
173 C Mathematical constants
174 C-----
175 C      DATA ONE, HALF, TWELVE, TWO, ZERO/1.0E0, 0.5E0, 12.0E0, 2.0E0, 0.0E0/,
176 C      1      SQRTPI/0.9189385332046727417803297E0/,
177 C      2      PI/3.1415926535897932384626434E0/
178 CD     DATA ONE, HALF, TWELVE, TWO, ZERO/1.0D0, 0.5D0, 12.0D0, 2.0D0, 0.0D0/,
179 CD     1      SQRTPI/0.9189385332046727417803297D0/,
180 CD     2      PI/3.1415926535897932384626434D0/
181 C-----
182 C Machine dependent parameters
183 C-----
184 C      DATA XBIG, XMININ, EPS/35.040E0, 1.18E-38, 1.19E-7/,
185 C      1      XINF/3.4E38/
186 CD     DATA XBIG, XMININ, EPS/171.624D0, 2.23D-308, 2.22D-16/,
187 CD     1      XINF/1.79D308/
188 C-----
189 C Numerator and denominator coefficients for rational minimax
190 C approximation over (1,2).
191 C-----
192 C      DATA P/-1.71618513886549492533811E+0, 2.47656508055759199108314E+1
193 C      1      -3.79804256470945635097577E+2, 6.29331155312818442661052E+2
194 C      2      8.66966202790413211295064E+2, -3.14512729688483675254357E+4
195 C      3      -3.61444134186911729807069E+4, 6.64561438202405440627855E+4
196 C      DATA Q/-3.08402300119738975254353E+1, 3.15350626979604161529144E+2
197 C      1      -1.01515636749021914166146E+3, -3.10777167157231109440444E+3
198 C      2      2.25381184209801510330112E+4, 4.75584627752788110767815E+3
199 C      3      -1.34659959864969306392456E+5, -1.15132259675553483497211E+5
200 CD     DATA P/-1.71618513886549492533811D+0, 2.47656508055759199108314D+1
201 CD     1      -3.79804256470945635097577D+2, 6.29331155312818442661052D+2
202 CD     2      8.66966202790413211295064D+2, -3.14512729688483675254357D+4
203 CD     3      -3.61444134186911729807069D+4, 6.64561438202405440627855D+4

```

^H

 PAGE 5
 04-12-91
 12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

204 CD DATA Q/-3.08402300119738975254353D+1,3.15350626979604161529144D+2
205 CD 1 -1.01515636749021914166146D+3,-3.10777167157231109440444D+3
206 CD 2 2.25381184209801510330112D+4,4.75584627752788110767815D+3
207 CD 3 -1.34659959864969306392456D+5,-1.15132259675553483497211D+5
208 C-----
209 C Coefficients for minimax approximation over (12, INF).
210 C-----
211 DATA C/-1.910444077728E-03,8.4171387781295E-04,
212 1 -5.952379913043012E-04,7.93650793500350248E-04,
213 2 -2.777777777777681622553E-03,8.3333333333333331554247E-02,
214 3 5.7083835261E-03/
215 CD DATA C/-1.910444077728D-03,8.4171387781295D-04,
216 CD 1 -5.952379913043012D-04,7.93650793500350248D-04,
217 CD 2 -2.777777777777681622553D-03,8.3333333333333331554247D-02,
218 CD 3 5.7083835261D-03/
219 C-----
220 C Statement functions for conversion between integer and float
221 C-----
222 CONV(I) = REAL(I)
223 CD CONV(I) = DBLE(I)
224 PARITY = .FALSE.
225 FACT = ONE
226 N = 0
227 Y = X
228 IF (Y .LE. ZERO) THEN
229 C-----
230 C Argument is negative
231 C-----
232 Y = -X
233 Y1 = AINT(Y)
234 RES = Y - Y1
235 IF (RES .NE. ZERO) THEN
236 IF (Y1 .NE. AINT(Y1*HALF)*TWO) PARITY = .TRUE.
237 FACT = -PI / SIN(PI*RES)
238 Y = Y + ONE
239 ELSE
240 RES = XINF
241 GO TO 900
242 END IF
243 END IF
244 C-----
245 C Argument is positive
246 C-----
247 IF (Y .LT. EPS) THEN
248 C-----
249 C Argument .LT. EPS
250 C-----
251 IF (Y .GE. XMININ) THEN
252 RES = ONE / Y
253 ELSE
254 RES = XINF
255 GO TO 900
256 END IF
257 ELSE IF (Y .LT. TWELVE) THEN
258 Y1 = Y
259 IF (Y .LT. ONE) THEN
260 C-----
261 C 0.0 .LT. argument .LT. 1.0

```


^H

PAGE
04-12-91
12:09:00

6

```

Line# Source Line      Microsoft FORTRAN Optimizing Compiler Version 5.00

262 C-----
263       Z = Y
264       Y = Y + ONE
265       ELSE
266 C-----
267 C 1.0 .LT. argument .LT. 12.0, reduce argument if necessary
268 C-----
269       N = INT(Y) - 1
270       Y = Y - CONV(N)
271       Z = Y - ONE
272       END IF
273 C-----
274 C Evaluate approximation for 1.0 .LT. argument .LT. 2.0
275 C-----
276       XNUM = ZERO
277       XDEN = ONE
278       DO 260 I = 1, 8
279           XNUM = (XNUM + P(I)) * Z
280           XDEN = XDEN * Z + Q(I)
281 260     CONTINUE
282       RES = XNUM / XDEN + ONE
283       IF (Y1 .LT. Y) THEN
284 C-----
285 C Adjust result for case 0.0 .LT. argument .LT. 1.0
286 C-----
287       RES = RES / Y1
288       ELSE IF (Y1 .GT. Y) THEN
289 C-----
290 C Adjust result for case 2.0 .LT. argument .LT. 12.0
291 C-----
292       DO 290 I = 1, N
293           RES = RES * Y
294           Y = Y + ONE
295 290     CONTINUE
296       END IF
297       ELSE
298 C-----
299 C Evaluate for argument .GE. 12.0,
300 C-----
301       IF (Y .LE. XBIG) THEN
302           YSQ = Y * Y
303           SUM = C(7)
304           DO 350 I = 1, 6
305               SUM = SUM / YSQ + C(I)
306 350     CONTINUE
307           SUM = SUM/Y - Y + SQRTPI
308           SUM = SUM + (Y-HALF)*LOG(Y)
309           RES = EXP(SUM)
310       ELSE
311           RES = XINF
312           GO TO 900
313       END IF
314     END IF
315 C-----
316 C Final adjustments and return
317 C-----
318       IF (PARITY) RES = -RES
319       IF (FACT .NE. ONE) RES = FACT / RES

```

^H

 PAGE 7
 04-12-91
 12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

320      900 GAMMA = RES
321      CD900 DGAMMA = RES
322      RETURN
323      C ----- Last line of GAMMA -----
324      END

```

GAMMA Local Symbols

Name	Class	Type	Size	Offset
GAMMA	param			0006
X	param			0008
XNUM.	local	REAL*4	4	004e
I	local	INTEGER*4	4	0052
Y1.	local	REAL*4	4	0056
N	local	INTEGER*4	4	005a
Y	local	REAL*4	4	005e
PARITY.	local	LOGICAL*4	4	0062
Z	local	REAL*4	4	0066
FACT.	local	REAL*4	4	006a
RES	local	REAL*4	4	006e
XDEN.	local	REAL*4	4	0072
SUM	local	REAL*4	4	0076
YSQ	local	REAL*4	4	007a
ONE	local	REAL*4	4	01bc
HALF.	local	REAL*4	4	01c0
TWELVE.	local	REAL*4	4	01c4
TWO	local	REAL*4	4	01c8
ZERO.	local	REAL*4	4	01cc
SQRTPI.	local	REAL*4	4	01d0
PI.	local	REAL*4	4	01d4
XBIG.	local	REAL*4	4	01d8
XMININ.	local	REAL*4	4	01dc
EPS	local	REAL*4	4	01e0
XINF.	local	REAL*4	4	01e4
P	local	REAL*4	32	01e8
Q	local	REAL*4	32	0208
C	local	REAL*4	28	0228

```

325      C----- calculate 1/B -----
326      SUBROUTINE FUNCT1 (B,E9)
327      IMPLICIT REAL (A-Z)
328      JI=.02
329      JF=2.1
330      10  J=JI
331      NI=FUNCT2(J)
332      J=JF
333      NF=FUNCT2(J)
334      A=(NF-NI)/(JF-JI)
335      B=NI-A*JI
336      NJ=FUNCT2(J)
337      IF ((NJ-E9).LT..001) GO TO 20
338      IF (NJ.GT.E9) THEN
339          JF=J
340      ELSE
341          JI=J

```


^H

 PAGE 8
 04-12-91
 12:09:00

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

342      END IF
343      GO TO 10
344  20    B=J
345      RETURN
346      END

```

FUNCT1 Local Symbols

Name	Class	Type	Size	Offset
E9.	param			0006
B	param			000a
A	local	REAL*4	4	007e
J	local	REAL*4	4	0082
JF.	local	REAL*4	4	0086
JI.	local	REAL*4	4	008a
NF.	local	REAL*4	4	008e
NI.	local	REAL*4	4	0092
NJ.	local	REAL*4	4	0096

```

347  C-----
348  C-----
350      FUNCTION FUNCT2 (J)
351      IMPLICIT REAL (A-Z)
352  C----- calculate n(j) -----
353      N=1+J
354      G1=GAMMA(N)
355      N=1+2*J
356      G2=GAMMA(N)
357      N=1+3*J
358      G3=GAMMA(N)
359      N1=G3-3*G2*G1+2*G1**3
360      N22=(G2-G1*G1)**1.5
361      FUNCT2=N1/N22
362      RETURN
363      END

```

FUNCT2 Local Symbols

Name	Class	Type	Size	Offset
FUNCT2.	param			0006
J	param			0008
N	local	REAL*4	4	009a
N22	local	REAL*4	4	009e
G1.	local	REAL*4	4	00a2
G2.	local	REAL*4	4	00a6
G3.	local	REAL*4	4	00aa
N1.	local	REAL*4	4	00ae

Global Symbols

Name	Class	Type	Size	Offset
FUNCT1.	FSUBRT	***	***	09bc

^H

PAGE 9
04-12-91
12:09:00

Microsoft FORTRAN Optimizing Compiler Version 5.00

Global Symbols

Name	Class	Type	Size	Offset
FUNCT2.	FFUNCT	REAL*4	***	0b1a
GAMMA	FFUNCT	REAL*4	***	059b
WEIBUL.	FSUBRT	***	***	0000

Code size = 0c45 (3141)

Data size = 0272 (626)

Bss size = 00b2 (178)

No errors detected

^H

 PAGE 1
 04-12-91
 12:08:04

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

1      SUBROUTINE GENSUR (NX,NY,X,Y,Z,XAPRIN,YMIN1,DYMIN1,YMIN2,DYMIN2,
2      1XMIN1,DXMIN1,XMIN2,DXMIN2,ZMIN,DZMIN,YPA1,NXEND,YPA2,NYSTAR,NYPOL
3      2XIPOL,XFPOL,NXSTAR,NXPOL,XPOL,ZPOL,YPOL,XCOL,YCOL,ZCOL,NCOLEF,
4      3NCORIG,NPTLEF,NPTRIG,ZSOL,NSOIL,NYEND,DIFF)
5      C.....this routine generates three dimensional surfaces
6      C.....It calls routines POLYGO and POLYGZ
7      $include:'CURRENTF.INC'
      ***** Begin listing of: CURRENTF.INC
1      *
2      *===== Prologue =====
3      *
4      * Purpose:
5      *      Include file with parameter declaration for the program PCSTL3D
6      *
7      * History:
8      *      Version          Programmer      Date          Description
9      *      -----          -
10     *      1.0              G.Moshkovich    ----          created
11     *
12     * IN args/commons          Units          Description
13     * -----          -
14     *
15     * OUT args/commons          Units          Description
16     * -----          -
17     *
18     * INOUT args/commons          Units          Description
19     * -----          -
20     *
21     * Processing:
22     *      none
23     *
24     * Special Requirements:
25     *      none
26     *
27     *----- Include files -----
28     *----- Constant declarations -----
29     integer MCOORD, MSOIL, NUMCOL, NCORN
30
31     parameter (MCOORD = 25, MSOIL = 5)
32     parameter (NUMCOL = 10, NCORN = 4)
33     PARAMETER (MAXSUR=2000)
34
35     *----- MCOORD - maximum number of points in each direction
36     *----- MSOIL - maximum number of soil layers
37     *----- NUMCOL - maximum number of generated columns
38     *----- NCORN - number of nods for each column (do not change)
39
40     *----- Argument declarations -----
41     *----- Global/External declarations -----
42     *----- Local declarations -----
43
44
45     ***** End listing of: CURRENTF.INC
8      DIMENSION X(MCOORD), Y(MCOORD), Z(MSOIL,MCOORD,MCOORD),
9      1XMIN1(MCOORD), DXMIN1(MCOORD), XMIN2(MCOORD), DXMIN2(MCOORD),
10     2ZMIN(MCOORD,MCOORD), DZMIN(MCOORD,MCOORD), YPOLAX(MCOORD),
11     3ZPOLAX(MCOORD), ZMIAX(MCOORD), DZMIAX(MCOORD), XIPOL(MCOORD),

```

^H

 PAGE 2
 04-12-91
 12:08:04

```

Line#  Source Line      Microsoft FORTRAN Optimizing Compiler Version 5.00

12      4XFPOL(MCOORD), XMIN3(MCOORD), DXMIN3(MCOORD), NXPOL(MCOORD),
13      5NXSTAR(MCOORD), NXEND(MCOORD), XPOL(MCOORD,MCOORD), YPOL(MCOORD),
14      6ZPOL(MCOORD,MCOORD), XDIFF(MCOORD), XPOL3(MCOORD), ZPOL3(MCOORD),
15      7XCOL(MCOORD,-NUMCOL:NUMCOL,NCORN), YCOL(MCOORD,-NUMCOL:NUMCOL,
16      8NCORN), ZCOL(MCOORD,-NUMCOL:NUMCOL,NCORN), NCOLEF(MCOORD),
17      9NPTLEF(MCOORD), NCORIG(MCOORD), NPTRIG(MCOORD), ZSOL(MSOIL,MCOORD
18      X-NUMCOL:NUMCOL,NCORN)
19      INTEGER XAPRIN,XROW,DIFF,XDIFF
20      CHARACTER CODE
21      REAL K1
22
23      C.....USE SYSTEM TIME AS SEED FOR RANDOM ROUTINS
24      CALL GETTIM(IHR,IMIN,ISEC,I100TH)
25      RANDC=(I100TH+0.5)/100
26      CALL GETTIM(IHR,IMIN,ISEC,I100TH)
27      RANDL=(I100TH+0.5)/100
28      CALL GETTIM(IHR,IMIN,ISEC,I100TH)
29      RANDR=(I100TH+0.5)/100
30      C.....GENERATES THE MAIN AXIS POLYGONAL.....
31      C.....first vertice of surface on Y direction.....
32      CALL RNDC (RANDC)
33      YPA1=YMIN1+RANDC*DYMIN1
34      YTESTE=YPA1
35      XROW=XAPRIN
36      CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
37      ZPA1=ZFINAL
38      K1=ALPHA
39      C.....last vertice of surface on Y direction.....
40      CALL RNDC (RANDC)
41      YPA2=YMIN2+RANDC*DYMIN2
42      YTESTE=YPA2
43      XROW=XAPRIN
44      CALL COORYZ (XROW,Y,Z,NY,YTESTE,ZFINAL,ALPHA)
45      ZPA2=ZFINAL
46      C.....calculation of number of vertices between extremities.....
47      DO 10 I=1,NY
48          IF (Y(I).GT.YPA1) THEN
49              NYSTAR=I
50              GO TO 20
51          END IF
52      10  CONTINUE
53      20  CONTINUE
54      DO 30 I=NY,1,-1
55          IF (Y(I).LT.YPA2) THEN
56              NYEND=I
57              GO TO 40
58          END IF
59      30  CONTINUE
60      40  CONTINUE
61      C.....nypol = number of Y vertices in the main axis polygonal.....
62      C.....including extremities.....
63      NYPOL=NYEND-NYSTAR+3
64      YPOLAX(1)=YPA1
65      ZPOLAX(1)=ZPA1
66      YPOLAX(NYPOL)=YPA2
67      ZPOLAX(NYPOL)=ZPA2
68      NYPPOL=NYPOL-1
69      DIFF=NYSTAR-2

```

^H

 PAGE 3
 04-12-91
 12:08:04

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

70 C.....transform all y coordinates to ypolax coordinates.....
71     DO 50 I=2,NYPPOL
72         J=I+DIFF
73         YPOLAX(I)=Y(J)
74     50 CONTINUE
75 C.....here zmin is transformed in an unidimensional array to fit the
76 C.....routine call internal dimension.....
77     DO 60 I=1,NY
78         ZMIAX(I)=ZMIN(XAPRIN,I)
79         DZMIAX(I)=DZMIN(XAPRIN,I)
80     60 CONTINUE
81 C.....call routine to random generate the coordinates.....
82 C.....code identifies which side is being generated (l=left,r=right,
83 C.....c=axis)
84     CODE='c'
85 C          {-----'input'-----}
86     CALL POLYGO (NYPPOL,YPOLAX,ZMIAX,DZMIAX,CODE,DIFF,K1,RANDC,RANDL,
87     1RANDR,XAPRIN,Y,Z,NY,ZPOLAX)
88 C          {output}
89 C.....NOW GENERATES THE LEFT SIDE INTERSECTION SURFACE/GROUND.....
90     CODE='l'
91 C.....these are the extremities of the intersection (on the main axis)
92     XIPOL(1)=X(XAPRIN)
93     XIPOL(NYPOL)=X(XAPRIN)
94 C          {-----'input'-----}
95     CALL POLYGO (NYPPOL,YPOLAX,XMIN1,DXMIN1,CODE,DIFF,K1,RANDC,RANDL,
96     1RANDR,XAPRIN,Y,Z,NY,XIPOL)
97 C          {output}
98 C.....NOW GENERATES THE RIGHT SIDE INTERSECTION SURFACE/GROUND.....
99     CODE='r'
100 C.....these are the extremities of the intersection (on the main axis)
101 C.....the signal is inverted to let the angle definition in the polygo
102 C.....nal routine be the same even though here the orientation of the
103 C.....boundaries are inversed.....
104     XFPOL(1)=-X(XAPRIN)
105     XFPOL(NYPOL)=-X(XAPRIN)
106 C.....now transform arrays xmin2 and dxmin2 so that routine can be
107 C.....used without changing orientation of angles.....
108     DO 70 I=1,NY
109         XMIN3(I)=-(XMIN2(I)+DXMIN2(I))
110         DXMIN3(I)=DXMIN2(I)
111     70 CONTINUE
112 C          {-----'input'-----}
113     CALL POLYGO (NYPPOL,YPOLAX,XMIN3,DXMIN3,CODE,DIFF,K1,RANDC,RANDL,
114     1RANDR,XAPRIN,Y,Z,NY,XFPOL)
115 C          {output}
116     DO 80 I=1,NYPOL
117         XFPOL(I)=-XFPOL(I)
118     80 CONTINUE
119 C.....NOW CALCULATE THE BODY OF THE SURFACE ITSELF.....
120 C.....calculate , for each Y coordinate , the amount of vertices.....
121 C.....in the X direction .....
122     NXPOL(1)=1
123     NXPOL(NYPOL)=1
124     DO 200 I=2,NYPPOL
125 C.....define limiting variables.....
126     J1=I+DIFF
127     XI=XIPOL(I)

```

^H

 PAGE 4
 04-12-91
 12:08:04

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

128          XF=XFPOL(I)
129 C.....calculate the number of points in the X direction for that Y...
130 C.....coordinate.....
131          DO 90 J=1,XAPRIN
132              IF (X(J).GT.XI) THEN
133                  NXSTAR(I)=J
134                  GO TO 100
135              END IF
136 90          CONTINUE
137 100         CONTINUE
138          DO 110 J=XAPRIN,NX
139              IF (X(J).GE.XF) THEN
140                  NXEND(I)=J-1
141                  GO TO 120
142              END IF
143 110         CONTINUE
144 120         CONTINUE
145          NXPOL(I)=NXEND(I)-NXSTAR(I)+3
146          XDIF(I)=NXSTAR(I)-2
147 C.....association of the polygonal relative X coordinates with the....
148 C.....absolute reference X axis.....
149 C.....xpol(il,i) is the coordinate X of the il-enth point of the surf.
150 C.....at the i-enth Y coordinate. il varies from 1 to nxpol(i) , and i
151 C.....varies from 1 to nypol
152          XPOL(1,I)=XI
153          XPOL(NXPOL(I),I)=XF
154          L=NXSTAR(I)
155          LIM=NXPOL(I)-1
156          DO 130 I1=2,LIM
157              XPOL(I1,I)=X(L)
158              L=L+1
159 130         CONTINUE
160 C.....calculate the Z values for the initial and final points.....
161          NYROW=I
162          XTESTE=XI
163          CALL COORXZ (NYROW,X,Z,NX,XTESTE,ZTESTE)
164          ZPOL(1,I)=ZTESTE
165 C          .....
166          XTESTE=XF
167          CALL COORXZ (NYROW,X,Z,NX,XTESTE,ZTESTE)
168          ZPOL(NXPOL(I),I)=ZTESTE
169 C.....Calculate the values of Z of polygonal on the LEFT.....
170 C.....transform array zmin to fit dimension of routine polygonal.
171 C.....At this part, the polygonal will be generated assuming that
172 C.....xpol3 and zmiac start at x = xaprin and the axis of X increases
173 C.....to the left . Thus zmiac(1) = zmin(xaprin) and ndiffx =0 (both
174 C.....start at xaprin ) .
175          L=XAPRIN
176          DO 140 I1=1,XAPRIN
177              ZMIAX(I1)=ZMIN(L,I)
178              DZMIAX(I1)=DZMIN(L,I)
179              L=L-1
180 140         CONTINUE
181          NXPA=XAPRIN-XDIF(I)
182          NXPAB=NXPOL(I)-1
183          ZPOL(NXPA,I)=ZPOLAX(I)
184 C.....no transform arrays xpol and zmiac on the routine call because.....
185 C.....used without changing orientation of angles.....

```


^H

 PAGE 5
 04-12-91
 12:08:04

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

186          K=NXPA
187          DO 150 J=1,NXPA
188              XPOL3(J)=-XPOL(K,I)
189              K=K-1
190      150      CONTINUE
191          ZPOL3(1)=ZPOLAX(I)
192      C.....angle is horizontal
193          K1=0
194          CODE='1'
195          NDIFFX=0
196          NYNOW=I
197      C          {-----'input'-----}
198          CALL POLYGO (NXPAB,XPOL3,ZMIAX,DZMIAX,CODE,NDIFFX,K1,RANDC,
199      1          RANDL,RANDR,NYNOW,X,Z,0,ZPOL3)
200      C          {output}
201      C.....restore array to original.....
202          K=NXPA-1
203          DO 160 J=2,NXPA-1
204              ZPOL(K,I)=ZPOL3(J)
205              K=K-1
206      160      CONTINUE
207      C.....Calculate values of Z on RIGHT side of polygonal.....
208      C.....no≠ transform array≤ xpo∞ so tha[ routing cae bo.....
209      C.....used without changing orientation of angles.....
210          I1=1
211          DO 170 I2=XAPRIN,NX
212              ZMIAX(I1)=ZMIN(I2,I)
213              DZMIAX(I1)=DZMIN(I2,I)
214              I1=I1+1
215      170      CONTINUE
216          NXPAB=NXPOL(I)-NXPA
217          J1=1
218          DO 180 J=NXPA,NXPOL(I)
219              XPOL3(J1)=XPOL(J,I)
220              J1=J1+1
221      180      CONTINUE
222          ZPOL3(1)=ZPOLAX(I)
223      C.....angle is horizontal
224          K1=0
225          CODE='r'
226          NDIFFX=0
227          NYNOW=I
228      C          {-----'input'-----}
229          CALL POLYGO (NXPAB,XPOL3,ZMIAX,DZMIAX,CODE,NDIFFX,K1,RANDC,
230      1          RANDL,RANDR,NYNOW,X,Z,0,ZPOL3)
231      C          {output}
232      C.....restore array to original.....
233          K=NXPA+1
234          DO 190 J=2,NXPAB
235              ZPOL(K,I)=ZPOL3(J)
236              K=K+1
237      190      CONTINUE
238      200      CONTINUE
239      C-----Transfer coordinates-----
240          CALL TRANSF (YPA1,YPA2,NYSTAR,NYPOL,Y,YPOL)
241      C          {-----input-----}{-out-}
242      C-----Calculate columns of the left side-----
243          CALL COLUML (YPA1,YPA2,NYSTAR,NYPOL,X,Y,Z,XAPRIN,XIPOL,NXSTAR,

```


^H

 PAGE 6
 04-12-91
 12:08:04

Line# Source Line Microsoft FORTRAN Optimizing Compiler Version 5.00

```

244      1NXPOL,XPOL,YPOL,XCOL,YCOL,ZCOL,NCOLEF,NPTLEF,ZPOL,ZSOL,XDIFF,
245      2NSOIL,NPPOL)
246  C-----Calculate columns of the right side-----
247      CALL COLUMR (YPA1,YPA2,NYSTAR,NYPOL,NPPOL,X,Y,Z,XAPRIN,XFPOL,
248      1NXSTAR,NXPOL,XPOL,YPOL,ZPOL,XCOL,YCOL,ZCOL,NCOLEF,NCORIG,NPTLEF,
249      2NPTRIG,NXEND,ZSOL,XDIFF,NSOIL)
250      RETURN
251      END

```

GENSUR Local Symbols

Name	Class	Type	Size	Offset
DIFF.	param			0006
NYEND	param			000a
NSOIL	param			000e
ZSOL.	param			0012
NPTRIG.	param			0016
NPTLEF.	param			001a
NCORIG.	param			001e
NCOLEF.	param			0022
ZCOL.	param			0026
YCOL.	param			002a
XCOL.	param			002e
YPOL.	param			0032
ZPOL.	param			0036
XPOL.	param			003a
NXPOL	param			003e
NXSTAR.	param			0042
XFPOL	param			0046
XIPOL	param			004a
NYPOL	param			004e
NYSTAR.	param			0052
YPA2.	param			0056
NXEND	param			005a
YPA1.	param			005e
DZMIN	param			0062
ZMIN.	param			0066
DXMIN2.	param			006a
XMIN2	param			006e
DXMIN1.	param			0072
XMIN1	param			0076
DYMIN2.	param			007a
YMIN2	param			007e
DYMIN1.	param			0082
YMIN1	param			0086
XAPRIN.	param			008a
Z	param			008e
Y	param			0092
X	param			0096
NY.	param			009a
NX.	param			009e
ZFINAL.	local	REAL*4	4	0000
NPPOL	local	INTEGER*4	4	0004
I	local	INTEGER*4	4	0008
ZMIAX	local	REAL*4	100	000c
J	local	INTEGER*4	4	0070

^H

PAGE 7
04-12-91
12:08:04

Microsoft FORTRAN Optimizing Compiler Version 5.00

GENSUR Local Symbols

Name	Class	Type	Size	Offset
K	local	INTEGER*4	4	0074
L	local	INTEGER*4	4	0078
DZMIAX.	local	REAL*4	100	007c
XROW.	local	INTEGER*4	4	00e0
NYNOW	local	INTEGER*4	4	00e4
CODE.	local	CHAR*1	1	00e8
ZPA1.	local	REAL*4	4	00ea
XTESTE.	local	REAL*4	4	00ee
ZPA2.	local	REAL*4	4	00f2
YPOLAX.	local	REAL*4	100	00f6
XF.	local	REAL*4	4	015a
YTESTE.	local	REAL*4	4	015e
ZPOLAX.	local	REAL*4	100	0162
ZTESTE.	local	REAL*4	4	01c6
NYROW	local	INTEGER*4	4	01ca
XI.	local	REAL*4	4	01ce
LIM	local	INTEGER*4	4	01d2
NYPPOL.	local	INTEGER*4	4	01d6
IHR	local	INTEGER*4	4	01da
ISEC.	local	INTEGER*4	4	01de
ALPHA	local	REAL*4	4	01e2
RANDC	local	REAL*4	4	01e6
IMIN.	local	INTEGER*4	4	01ea
XMIN3	local	REAL*4	100	01ee
RANDL	local	REAL*4	4	0252
XDIFF	local	INTEGER*4	100	0256
DXMIN3.	local	REAL*4	100	02ba
I100TH.	local	INTEGER*4	4	031e
XPOL3	local	REAL*4	100	0322
NXPA.	local	INTEGER*4	4	0386
RANDR	local	REAL*4	4	038a
ZPOL3	local	REAL*4	100	038e
NXPAB	local	INTEGER*4	4	03f2
I1.	local	INTEGER*4	4	03f6
I2.	local	INTEGER*4	4	03fa
J1.	local	INTEGER*4	4	03fe
K1.	local	REAL*4	4	0402
NDIFFX.	local	INTEGER*4	4	0406

Symbolic Constant	Type	Value
MCOORD.	INTEGER*4	25
MSOIL	INTEGER*4	5
NUMCOL.	INTEGER*4	10
NCORN	INTEGER*4	4
MAXSUR.	INTEGER*4	2000

^H

 PAGE 8
 04-12-91
 12:08:04

Microsoft FORTRAN Optimizing Compiler Version 5.00

Global Symbols

Name	Class	Type	Size	Offset
COLUML.	extern	***	***	***
COLUMR.	extern	***	***	***
COORXZ.	extern	***	***	***
COORYZ.	extern	***	***	***
GENSUR.	FSUBRT	***	***	0000
GETTIM.	extern	***	***	***
POLYGO.	extern	***	***	***
RNDC.	extern	***	***	***
TRANSF.	extern	***	***	***

Code size = 13ac (5036)

Data size = 000d (13)

Bss size = 040a (1034)

No errors detected

PRE/POSTPROCESSOR

```

DECLARE SUB PopDira (SEARCH$, SHOWITEMS%, FORE%, BACK%, HFORE%, HBACK%, QUADRANTS$,
SHADOW%, NUMFILES%, SELECTFILES$, RETURN.CODE%)
DECLARE SUB NewGetIn (FldContent AS STRING, numonly%, MaxLen%, FldRow%, LeftColumn%,
ColorAttr%, ReturnVar AS STRING, AbortVar AS STRING)
'$INCLUDE: 'CURRENTB.BI'

```

```

CLEAR , , 7000

```

```

ON ERROR GOTO LocalHandler

```

```

KEY 15, CHR$(4) + CHR$(&H4F)
KEY 16, CHR$(132) + CHR$(&H4F)
ON KEY(15) GOSUB GotEndKey
ON KEY(16) GOSUB GotEndKey

```

```

Headline1
Delay 2
ManeMenu

```

```

END

```

```

LocalHandler:
    GlobalHandler
    RESUME NEXT

```

```

GotEndKey:

```

```

    REDIM ARRY%(2000)
    SCRNNARY.SEGMENT% = VARSEG(ARRY%(0))
    SCRNNARY.OFFSET% = VARPTR(ARRY%(0))
    CALL SAVESCRN(-1, SCRNNARY.SEGMENT%, SCRNNARY.OFFSET%, RETURN.CODE%)
    Title$ = "Please choose desired option"
    Foreground% = 4: Background% = 7
    List$(0) = "Cancel"
    List$(1) = "Exit and save changes"
    List$(2) = "Exit and don't save changes"
    CALL POPMENU(Title$, 3, List$(), 2, Foreground%, Background%, Foreground%,
Background%, "0", 1, Choice%, RETURN.CODE%)
    SCRNNARY.SEGMENT% = VARSEG(ARRY%(0))
    SCRNNARY.OFFSET% = VARPTR(ARRY%(0))
    CALL RESTSCRN(-1, SCRNNARY.SEGMENT%, SCRNNARY.OFFSET%, RETURN.CODE%)
    ERASE ARRY%

```

```

SELECT CASE Choice%
CASE 1
    RETURN

```

```

CASE 2
    PrintData ExitCode
    KEY(15) OFF
    KEY(16) OFF
    ClearVar
    ManeMenu

```

```

CASE 3
    TextLine$(1) = "You are about to loose all your edition"
    TextLine$(2) = "Please press 'y' to confirm this operation"
    TextLine$(3) = "Any other key will resume your current operation "

    AskWindow 3, TextLine$(), KeyPressed$
    IF KeyPressed$ = "y" OR KeyPressed$ = "Y" THEN
        KEY(15) OFF
        KEY(16) OFF
        ClearVar
        ManeMenu
    ELSE
        RETURN
    END IF

```

```

CASE ELSE
    RETURN
END SELECT

REM $DYNAMIC
SUB ClearScrn
    CLS
    FOR I = 1 TO 25
        LOCATE I, 1: PRINT STRING$(80, CHR$(177));
    NEXT I
END SUB

SUB Delay (delt)
    LoopFin = TIMER + delt
    DO WHILE TIMER < LoopFin
        LOOP
    END SUB

SUB Footer
    COLOR 0, 0
    CLS
    CALL MAKEWIND(23, 4, 24, 76, 2, 1, 11, 0, 0, "", RETURN.CODE%)
    COLOR 1, 11
    LOCATE 23, 15: PRINT "PC-STABL3D - 3 Dimensional Slope Stability Program";
    LOCATE 24, 15: PRINT "                        Version 2.2";
    COLOR 0, 0
END SUB

SUB Headline
    SCREEN 0, 1, 0, 0
    WIDTH 80
    COLOR 1, 7
    ClearScrn
    CALL MAKEWIND(23, 4, 24, 76, 2, 1, 11, 0, 0, "", RETURN.CODE%)
    COLOR 1, 11
    LOCATE 23, 15, 0: PRINT "PC-STABL3D - 3 Dimensional Slope Stability Program"
    LOCATE 24, 15, 0: PRINT "                        Version 2.2 ";
END SUB

SUB Headline1
    SCREEN 0, 1, 0, 0
    WIDTH 80, 25
    COLOR 1, 7
    ClearScrn
    CALL MAKEWIND(9, 18, 16, 65, 2, 0, 7, 1, 0, "", RETURN.CODE%)
    COLOR 0, 7
    LOCATE 9, 22, 0: PRINT " PC-STABL3D - SLOPE STABILITY PROGRAM ";
    LOCATE 10, 22, 0: PRINT "      GRAPHICAL PRE/POST PROCESSOR      ";
    LOCATE 12, 22, 0: PRINT "      Civil Engineering Department      ";
    LOCATE 13, 22, 0: PRINT "      Purdue University      ";
    LOCATE 15, 22, 0: PRINT "Version 2.1 by J. Eduardo Thomaz, 1987";
    LOCATE 16, 22, 0: PRINT " Version 2.2 by G. Moshkovich, 1990 ";
END SUB

SUB ManeMenu
    STATIC ExitCode AS INTEGER
    DO
        Headline

        '-----
        '
        '                                Main Menu
        '-----

        List$(0) = "      Create Data File      "
        List$(1) = "      Edit Existing Data File      "

```

```
List$(2) = "    Read  Data File          "
List$(3) = "    Plot 3D Geometry        "
List$(4) = "    Plot Cross Sections     "
List$(5) = "    Run PCSTBL3D             "
List$(6) = "    Plot 3D Surface Contours  "
List$(7) = "    Plot Surface Cross Section "
List$(8) = "    Exit Program              "
```

```
Title$ = "    Main Menu    "
Foreground% = 0: Background% = 7
```

```
CALL POPMENU(Title$, 9, List$(), 2, 0, 7, Foreground%, Background%, "0", 1,
Choice%, RETURN.CODE%)
```

```
SELECT CASE Choice%
```

```
    CASE 1
```

```
        ClearVar
        InputData ExitCode
```

```
    CASE 2
```

```
        Caller$ = "edit"
        ClearVar
        InFile ExitCode
        IF ExitCode <> 666 THEN
            InputData ExitCode
        END IF
```

```
    CASE 3
```

```
        Caller$ = "geometry"
        FileIn% = 1
        InFile ExitCode
```

```
    CASE 4
```

```
        Contour
```

```
    CASE 5
```

```
        PlotCros
```

```
    CASE 6
```

```
        SCREEN 0
        COLOR 7, 0
        CLS
        RUN "pcstbl3d"
```

```
    CASE 7
```

```
        ReadCritFile ExitCode
        IF ExitCode <> 666 THEN
            Caller$ = "3dsurf"
            Contour
        END IF
```

```
    CASE 8
```

```
        IF FileIn% <> 1 THEN
            ERROR 243
        ELSE
            ReadCritFile ExitCode
            IF ExitCode <> 666 THEN
                PlotSurCr
            END IF
        END IF
```

```
    CASE 9
```

```
        ITEMS$(1) = " You are about to quit the program "
        ITEMS$(2) = " ??? DO YOU WANT TO DO IT ??? "
        ITEMS$(3) = "Press Y to confirm"
        AskWindow 3, ITEMS$(), SELECTION$
        IF UCASE$(SELECTION$) = "Y" THEN
            CLOSE
```



```

        COLOR 7, 0
        CLS
        PRINT "Version 2.1 by J.A. Thomas"
        PRINT "Version 2.2 by G. Moshkovich"
        END
    ELSE
        LOCATE 14, 2, 0: PRINT "
    END IF

CASE ELSE
END SELECT

LOOP
END SUB

SUB PlotCros

    Headline
    COLOR 0, 1
    CALL FASTPRT("
10, 7, 49, RETURN.CODE%)
    CALL FASTPRT("    Enter the number of the X cross section you want to check. ",
11, 7, 49, RETURN.CODE%)
    CALL FASTPRT("    ( remember you want the 'N' of X(N) ....) ",
12, 7, 49, RETURN.CODE%)
    CALL FASTPRT("    Enter N: ",
13, 7, 49, RETURN.CODE%)
    CALL FASTPRT("
14, 7, 49, RETURN.CODE%)

    CALL NewGetIn(" ", 0, 2, 13, 18, 10, ncro$, ab$)
    IF ab$ = CHR$(ESC) THEN EXIT SUB
    ncro = VAL(ncro$)

SCREEN 9
LOCATE 2, 24: PRINT "This is the cross section of x("; ncro; ")";
CALL Scale(ncro)

FOR K = 1 TO ns
    FOR J = 1 TO ny - 1
        LINE (y(J), z(ncro, J, K))-(y(J + 1), z(ncro, J + 1, K)), 12
    NEXT J
NEXT K

FOR K = 1 TO ns
    IF utype(K) <> 1 THEN
        FOR J = 1 TO ny - 1
            LINE (y(J), uz(ncro, J, K))-(y(J + 1), uz(ncro, J + 1, K))
13
        NEXT J
    END IF
NEXT K

' plot the boundary conditions
    FOR J = 1 TO ny
        LINE (y(J), zmin(ncro, J))-(y(J), zmin(ncro, J) + dzmin(ncro, J)), 2
    NEXT J

' plot limits on the main axis
    LeftStart = 0
    RightStart = 0
    FOR J = 1 TO ny
        SELECT CASE y(J)
            CASE yamin TO yamin + dyamin
                IF LeftStart = 0 THEN
                    LeftStart = 1
                    IndLeftStart = J
                END IF
                IndLeftFinish = J

```

```

CASE yamax TO yamax + dyamax
  IF RightStart = 0 THEN
    RightStart = 1
    IndRightStart = J
  END IF
  IndRightFinish = J

END SELECT
NEXT J
'calculate vertical coordinates of limiting points
ZLeftStart = (z(xaprin, IndLeftStart, 1) - z(xaprin, IndLeftStart - 1, 1)) /
(y(IndLeftStart) - y(IndLeftStart - 1)) * (yamin - y(IndLeftStart - 1)) + z(xaprin,
IndLeftStart - 1, 1)
ZLeftFinish = (z(xaprin, IndLeftFinish + 1, 1) - z(xaprin, IndLeftFinish, 1)) /
(y(IndLeftFinish + 1) - y(IndLeftFinish)) * (yamin + dyamin - y(IndLeftFinish)) +
z(xaprin, IndLeftFinish, 1)
ZRightStart = (z(xaprin, IndRightStart, 1) - z(xaprin, IndRightStart - 1, 1)) /
(y(IndRightStart) - y(IndRightStart - 1)) * (yamax - y(IndRightStart - 1)) +
z(xaprin, IndRightStart - 1, 1)
ZRightFinish = (z(xaprin, IndRightFinish + 1, 1) - z(xaprin, IndRightFinish,
1)) / (y(IndRightFinish + 1) - y(IndRightFinish)) * (yamax + dyamax -
y(IndRightFinish)) + z(xaprin, IndRightFinish, 1)

'plot lines at left and right boundaries
LINE (yamin, ZLeftStart)-(yamin, ZLeftStart + 4)
LINE (yamin + dyamin, ZLeftFinish)-(yamin + dyamin, ZLeftFinish + 4)
LINE (yamax, ZRightStart)-(yamax, ZRightStart + 4)
LINE (yamax + dyamax, ZRightFinish)-(yamax + dyamax, ZRightFinish + 4)

LOCATE 25, 22: PRINT "Press any key to go back to main menu ...";
SUSPEND$ = INPUT$(1)

END SUB

SUB PlotSurCr

  Headline
  COLOR 0, 1
  CALL FASTPRT("Enter the number of the X cross section you want to check. ",
10, 7, 49, RETURN.CODE%)
  CALL FASTPRT(" ( remember you want the 'N' of X(N) ....) ",
11, 7, 49, RETURN.CODE%)
  CALL FASTPRT("Enter N: ",
12, 7, 49, RETURN.CODE%)
  CALL FASTPRT(" ",
13, 7, 49, RETURN.CODE%)
  CALL FASTPRT(" ",
14, 7, 49, RETURN.CODE%)

  CALL NewGetIn(" ", 0, 2, 13, 18, 10, ncro$, ab$)
  IF ab$ = CHR$(ESC) THEN EXIT SUB
  ncro = VAL(ncro$) ' value of cross section

  SCREEN 9

  LOCATE 2, 24: PRINT "This is the cross section of x("; ncro; ")";

  CALL Scale(ncro)

  'WINDOW (y(0), y(0))-(y(ny - 1), y(ny - 1))
  'VIEW (51, 51)-(410, 300), 1, 11

  ' plot the critical surface
  FOR K = 1 TO ny - 1
    LINE (ysur(K), zsur(ncro, K))-(ysur(K + 1), zsur(ncro, K + 1)), 14
  NEXT K

  ' plot the piezometric line
  FOR K = 1 TO ns

```

```

      IF utype(K) <> 1 THEN
      FOR J = 1 TO ny - 1
        LINE (y(J), uz(ncro, J, K))-(y(J + 1), uz(ncro, J + 1, K))
11    NEXT J
      END IF
    NEXT K

    ' plot the soil profile
    FOR K = 1 TO ns
      FOR J = 1 TO ny - 1
        LINE (y(J), z(ncro, J, K))-(y(J + 1), z(ncro, J + 1, K)), 13
      NEXT J
    NEXT K

    ' plot the boundary conditions
    FOR J = 1 TO ny
      LINE (y(J), zmin(ncro, J))-(y(J), zmin(ncro, J) + dzmin(ncro, J)), 2
    NEXT J

    ' plot limits on the main axis
    LeftStart = 0
    RightStart = 0
    FOR J = 1 TO ny
      SELECT CASE y(J)
        CASE yamin TO yamin + dyamin
          IF LeftStart = 0 THEN
            LeftStart = 1
            IndLeftStart = J
          END IF
          IndLeftFinish = J

        CASE yamax TO yamax + dyamax
          IF RightStart = 0 THEN
            RightStart = 1
            IndRightStart = J
          END IF
          IndRightFinish = J
      END SELECT
    NEXT J

    'calculate vertical coordinates of limiting points
    ZLeftStart = (z(xaprin, IndLeftStart, 1) - z(xaprin, IndLeftStart - 1, 1)) /
(y(IndLeftStart) - y(IndLeftStart - 1)) * (yamin - y(IndLeftStart - 1)) + z(xaprin,
IndLeftStart - 1, 1)
    ZLeftFinish = (z(xaprin, IndLeftFinish + 1, 1) - z(xaprin, IndLeftFinish, 1)) /
(y(IndLeftFinish + 1) - y(IndLeftFinish)) * (yamin + dyamin - y(IndLeftFinish)) +
z(xaprin, IndLeftFinish, 1)
    ZRightStart = (z(xaprin, IndRightStart, 1) - z(xaprin, IndRightStart - 1, 1)) /
(y(IndRightStart) - y(IndRightStart - 1)) * (yamax - y(IndRightStart - 1)) +
z(xaprin, IndRightStart - 1, 1)
    ZRightFinish = (z(xaprin, IndRightFinish + 1, 1) - z(xaprin, IndRightFinish,
1)) / (y(IndRightFinish + 1) - y(IndRightFinish)) * (yamax + dyamax -
y(IndRightFinish)) + z(xaprin, IndRightFinish, 1)

    'plot lines at left and right boundaries
    LINE (yamin, ZLeftStart)-(yamin, ZLeftStart + 4)
    LINE (yamin + dyamin, ZLeftFinish)-(yamin + dyamin, ZLeftFinish + 4)
    LINE (yamax, ZRightStart)-(yamax, ZRightStart + 4)
    LINE (yamax + dyamax, ZRightFinish)-(yamax + dyamax, ZRightFinish + 4)

    LOCATE 25, 22: PRINT "Press any key to go back to main menu...";
    SUSPEND$ = INPUT$(1)

  END SUB

SUB PrVal (nc, z1(), K, colore)
LOCATE K + 3, 60: COLOR colore: PRINT USING "####.###"; z1(K)
LOCATE K + 3, 70: COLOR colore: PRINT "====="

```

END SUB

REM \$STATIC

SUB ReadCritFile (ExitCode AS INTEGER)

' This subroutine reads file with most critical surface data

Footer

CALL MAKEWIND(2, 3, 18, 78, 2, 11, 1, 1, 0, "", RETURN.CODE%)

CALL MAKEWIND(4, LCOL - 2, 5, RCOL + 2, 2, 1, 11, NOGROW, NOSHADOW, "",
RETURN.CODE%)

CALL FASTPRT("Select the name of the file with the 3D surface you want", 4, LCOL,
49, RETURN.CODE%)

CALL FASTPRT("to be plotted. Use the cursor keys and press <enter> ", 5, LCOL,
49, RETURN.CODE%)

DO

CALL PopDira("*.*", 6, 0, 11, 0, 11, "0", 1, numf%, File\$, RETURN.CODE%)

IF File\$ = "" OR INKEY\$ = CHR\$(ESC) THEN

ExitCode = 666

EXIT SUB

END IF

OPEN File\$ FOR INPUT AS #2

LINE INPUT #2, Legal\$

IF Legal\$ <> " PCSTBL3D v2.0 Critical Surface" THEN

ERROR 248

ErrorCode = 666

CLOSE

ELSE

ErrorCode = 999

END IF

LOOP UNTIL ErrorCode <> 666

COLOR 2, 1

CALL FASTPRT("|| >>> READING INPUT FILE DATA <<< >>> PLEASE WAIT <<< ||",
4, 9, 49, RETURN.CODE%)

CALL FASTPRT("|| ||",
5, 9, 49, RETURN.CODE%)

INPUT #2, nx, ny

' *** enter X coordinates ***

FOR J = 1 TO ny

FOR I = 1 TO nx

INPUT #2, xsur(I), ysur(J), zsur(I, J)

NEXT I

NEXT J

CLOSE

CALL FASTPRT("|| >>> FINISHED READING FILE <<< PLEASE PRESS ANY KEY ! ... ||",
4, 9, 49, RETURN.CODE%)

CALL FASTPRT("|| ||",
5, 9, 49, RETURN.CODE%)

SUSPEND\$ = INPUT\$(1)

ClearScrn

END SUB

SUB Scale (ncros)

' it calculates the values for the automatic scaling-----

IF Caller\$ = "geometry" THEN

```

ypmin = y(1)
ypmax = y(ny)
zpmin = 1000000000#
zpmax = -1000000000#

```

'.....sort for maximum and minimum values of the soil polygons.....

```

FOR I = 1 TO ns
  FOR J = 1 TO ny
    IF zpmax <= z(ncros, J, I) THEN zpmax = z(ncros, J, I)
    IF zpmin >= z(ncros, J, I) THEN zpmin = z(ncros, J, I)
  NEXT J
NEXT I

```

'.....sort also the limits for surface generation.....

```

FOR I = 1 TO nx
  FOR J = 1 TO ny
    IF zpmin > zmin(I, J) THEN zpmin = zmin(I, J)
  NEXT J
NEXT I

```

ELSE

```

ypmin = ysur(1)
ypmax = ysur(ny)
zpmin = 1000000000#
zpmax = -1000000000#

```

'.....sort for maximum and minimum values of the soil polygons.....

```

FOR J = 1 TO ny
  IF zpmax <= zsur(ncros, J) THEN zpmax = zsur(ncros, J)
  IF zpmin >= zsur(ncros, J) THEN zpmin = zsur(ncros, J)
NEXT J

```

END IF

'.....calculation of parameters for the graphics window.....

```

deltay = zpmax - zpmin
deltax = ypmax - ypmin
ymini = ypmin
ymaxi = ypmax
zmini = CINT(zpmin - (.74 * deltax - deltay) * .5)
zmaxi = CINT(zpmax + (.74 * deltax - deltay) * .5)

```

'.....calling the graphics routines.....

```

'COLOR 14, 1
'VIEW (53, 20)-(587, 329), 8, 2
VIEW (86, 40)-(552, 310), 8, 11
WINDOW (ymini, zmini)-(ymaxi, zmaxi)

```

```

LOCATE 24, 10: PRINT STR$(ymini);
LOCATE 24, 35: PRINT "Y A X I S";
LOCATE 24, 68: PRINT STR$(ymaxi);

```

```

LOCATE 3, 6: PRINT STR$(zmaxi);
LOCATE 10, 4: PRINT "Z";
LOCATE 12, 4: PRINT "A";
LOCATE 13, 4: PRINT "X";
LOCATE 14, 4: PRINT "I";
LOCATE 15, 4: PRINT "S";
LOCATE 23, 6: PRINT STR$(zmini);

```

END SUB

SUB VecOut (xstart, ystart, xstop, ystop, colore) STATIC

```
      LINE (xstart, ystart)-(xstop, ystop), colore  
END SUB
```

```

DECLARE SUB NewGetIn (FldContent AS STRING, numonly%, MaxLen%, FldRow%, LeftColumn%,
ColorAttr%, ReturnVar AS STRING, AbortVar AS STRING)
'$INCLUDE: 'CURRENTB.BI'

DATA 0,0,8,0,2,5,7,6,9,0,3,4,1,3,1,4,3,0,9,6,7,5,2,0,8,0,0
END

LocalHandler:
    GlobalHandler
    RESUME NEXT

REM $DYNAMIC
SUB Contour
ON ERROR GOTO LocalHandler
REM input variables to CONREC
REM d(0:iub, 0:jub) 'matrix for the data surface
REM iub,jub        'index bounds of the data array
REM xl(0:iub)       'data array for columns coordinates
REM yl(0:jub)       'data array for row coordinates
REM nc              'number of contour levels
REM xl(0:nc-1)      'contour levels in increasing order
REM vecout          'an external subroutine to plot the contour lines
REM false and true Boolean values

CALL Footer

IF Caller$ = "geometry" THEN

    CALL MAKEWIND(2, 3, 18, 78, 2, 11, 1, 0, 0, "", RETURN.CODE%)
    CALL FASTPRT(" ", 10, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| Enter the number of the soil layer you want to have the ||", 11, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| contours plotted: ||", 12, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| Soil layer number =      Enter number of contour levels: ||", 13, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| (number of levels <= 14) ||", 14, 9, 49, RETURN.CODE%)
    CALL FASTPRT(" ", 15, 9, 49, RETURN.CODE%)

    CALL NewGetIn("...", 0, 2, 13, 32, 10, soil$, ab$)
    IF ab$ = CHR$(ESC) THEN EXIT SUB
    soil = VAL(soil$)
ELSE

    CALL MAKEWIND(2, 3, 18, 78, 2, 11, 1, 0, 0, "", RETURN.CODE%)
    CALL FASTPRT(" ", 10, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| You chose to plot the contours of the most critical surface ||", 11, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| in your geometry: ||", 12, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| Now please enter the number of contour levels: ||", 13, 9, 49, RETURN.CODE%)
    CALL FASTPRT("|| (number of levels <= 14) ||", 14, 9, 49, RETURN.CODE%)
    CALL FASTPRT(" ", 15, 9, 49, RETURN.CODE%)

END IF

loop1:

    CALL NewGetIn("...", 0, 2, 13, 67, 10, nc$, ab$)
    IF ab$ = CHR$(ESC) THEN EXIT SUB
    nc = VAL(nc$)
    IF nc > 14 GOTO loop1

```



```

RESTORE
iub = nx - 1: jub = ny - 1
DIM d(iub + 1, jub + 1), z1(nc)

IF Caller$ = "geometry" THEN
  xmin = x(1): xmax = x(nx): ymin = y(1): ymax = y(ny)
ELSE
  xmin = xsur(1): xmax = xsur(nx): ymin = ysur(1): ymax = ysur(ny)
END IF

SCREEN 9
  'SCALE THE FIGURE
  deltax = xmax - xmin
  deltay = ymax - ymin
  xmini = xmin
  xmaxi = xmax
  ymini = CINT(ymin - (.89 * deltax - deltay) * .5)
  ymaxi = CINT(ymax + (.89 * deltax - deltay) * .5)
  WINDOW (xmini, ymini)-(xmaxi, ymaxi)
  VIEW (51, 51)-(410, 300), 0, 11

LOCATE 23, 6: PRINT STR$(xmin);
LOCATE 24, 25: PRINT "X AXIS";
LOCATE 23, 50: PRINT STR$(xmax);
LOCATE 4, 2: PRINT STR$(ymaxi);
LOCATE 10, 2: PRINT "Y"; : LOCATE 12, 2: PRINT "A";
LOCATE 13, 2: PRINT "X";
LOCATE 14, 2: PRINT "I";
LOCATE 15, 2: PRINT "S";
LOCATE 22, 2: PRINT STR$(ymini);

dmin = 1000000000#
dmax = -10000000000#

' ***** format data for geometry plot *****

IF Caller$ = "geometry" THEN

  FOR I = 1 TO nx
    FOR J = 1 TO ny
      x1(I) = x(I)
      y1(J) = y(J)
      d(I, J) = z(I, J, soil)
      IF d(I, J) <= dmin THEN dmin = d(I, J)
      IF d(I, J) >= dmax THEN dmax = d(I, J)
    NEXT J
  NEXT I

ELSEIF Caller$ = "3dsurf" THEN

  FOR I = 1 TO nx
    FOR J = 1 TO ny
      x1(I) = xsur(I)
      y1(I) = ysur(I)
      d(I, J) = zsur(I, J)
      IF d(I, J) <= dmin THEN dmin = d(I, J)
      IF d(I, J) >= dmax THEN dmax = d(I, J)
    NEXT J
  NEXT I

END IF

LOCATE 2, 60: PRINT "CONTOUR LEVELS"
FOR I = 1 TO nc
  z1(I) = dmin + I * (dmax - dmin) / nc
  CALL PrVal(nc, z1(), I, I)
NEXT I

```

```
' ***** Select the palette of colors to be used on the contours *****
```

```
    PALETTE 8, 11: PALETTE 7, 5: PALETTE 6, 13: PALETTE 10, 3
    PALETTE 5, 25: PALETTE 4, 9: PALETTE 3, 17: PALETTE 2, 1
    PALETTE 1, 8: PALETTE 9, 15: PALETTE 11, 19: PALETTE 12, 44
    PALETTE 13, 46: PALETTE 14, 55
```

```
GOSUB conrec
```

```
COLOR 7, 0
```

```
LOCATE 2, 9: PRINT "Press any key to go back to main menu ..."
```

```
SUSPEND$ = INPUT$(1)
```

```
EXIT SUB
```

```
END
```

```
' -----
conrec:
```

```
' local declarations for conrec
```

```
DIM h(5)      'relative heights of the box above contour
```

```
DIM ish(5)    'sign of h()
```

```
DIM xh(5)     'x coordinate of box
```

```
DIM yh(5)     'y coordinate of box
```

```
DIM im(4)     'mapping from vertex numbers to x offsets
```

```
DIM castab(3, 3, 3) 'case switch table
```

```
DIM jm(4)     'mapping from vertex numbers to offsets
```

```
im(1) = 0: im(2) = 1: im(3) = 1: im(4) = 0
```

```
jm(1) = 0: jm(2) = 0: jm(3) = 1: jm(4) = 1
```

```
FOR K% = 1 TO 3
```

```
  FOR J% = 1 TO 3
```

```
    FOR I% = 1 TO 3
```

```
      READ castab(K%, J%, I%)
```

```
    NEXT I%
```

```
  NEXT J%
```

```
NEXT K%
```

```
'CHECK THE INPUT PARAMETERS FOR VALIDITY
```

```
prmerr = FALSE
```

```
IF (iub <= 1 OR jub <= 1) THEN prmerr = TRUE
```

```
IF (nc <= 1) THEN prmerr = TRUE
```

```
FOR K = 2 TO nc
```

```
  IF (z1(K) <= z1(K - 1)) THEN prmerr = TRUE
```

```
NEXT K
```

```
IF (prmerr) THEN msg$ = " Error in input parameters ": RETURN
```

```
' SCAN THE ARRAY, TOP DOWN, LEFT TO RIGHT
```

```
FOR J = jub TO 1 STEP -1
```

```
FOR I = 1 TO iub
```

```
' find lowest vertex
```

```
IF (d(I, J) < d(I, J + 1)) THEN dmin = d(I, J) ELSE dmin = d(I, J + 1)
```

```
IF (d(I + 1, J) < dmin) THEN dmin = d(I + 1, J)
```

```
IF (d(I + 1, J + 1) < dmin) THEN dmin = d(I + 1, J + 1)
```

```
' find highest vertex
```

```
IF (d(I, J) > d(I, J + 1)) THEN dmax = d(I, J) ELSE dmax = d(I, J + 1)
```

```
IF (d(I + 1, J) > dmax) THEN dmax = d(I + 1, J)
```

```
IF (d(I + 1, J + 1) > dmax) THEN dmax = d(I + 1, J + 1)
```

```
IF (dmax < z1(1) OR dmin > z1(nc)) THEN GOTO noneinbox
```

```
' DRAW EACH CONTOUR WITHIN THIS BOX
```

```
FOR K = 1 TO nc
```

```
IF (z1(K) < dmin OR z1(K) > dmax) THEN GOTO noneintri
```

```
FOR M = 5 TO 1 STEP -1
```

```
  IF M > 1 THEN
    h(M) = d(I + im(M - 1), J + jm(M - 1)) - z1(K)
    xh(M) = x1(I + im(M - 1))
    yh(M) = y1(J + jm(M - 1))
  END IF
```

```
  IF M = 1 THEN
    h(1) = (h(2) + h(3) + h(4) + h(5)) / 4
    xh(1) = (x1(I) + x1(I + 1)) / 2
    yh(1) = (y1(J) + y1(J + 1)) / 2
  END IF
```

```
  IF h(M) > 0 THEN
    ish(M) = 3
  ELSEIF (h(M) < 0) THEN
    ish(M) = 1
  ELSE
    ish(M) = 2
  END IF
```

```
NEXT M
```

```
' SCAN EACH TRIANGLE IN THE BOX
```

```
FOR M = 2 TO 5
  m1 = M: m2 = 1: m3 = M + 1
  IF m3 = 6 THEN m3 = 2
  casa = CINT(castab(ish(m1), ish(m2), ish(m3)))
  IF casa = 0 THEN GOTO case0
  ON casa GOTO case1, case2, case3, case4, case5, case6, case7, case8, case9
```

```
'line between vertices m1 and m2
```

```
case1:
  x1 = xh(m1): y1 = yh(m1): x2 = xh(m2): y2 = yh(m2)
  GOTO drawit
```

```
case2:
  x1 = xh(m2): y1 = yh(m2): x2 = xh(m3): y2 = yh(m3)
  GOTO drawit
```

```
case3:
  x1 = xh(m3): y1 = yh(m3): x2 = xh(m1): y2 = yh(m1)
  GOTO drawit
```

```
case4:
  x1 = xh(m1): y1 = yh(m1)
  x2 = (h(m3) * xh(m2) - h(m2) * xh(m3)) / (h(m3) - h(m2))
  y2 = (h(m3) * yh(m2) - h(m2) * yh(m3)) / (h(m3) - h(m2))
  GOTO drawit
```

```
case5:
  x1 = xh(m2): y1 = yh(m2)
  x2 = (h(m1) * xh(m3) - h(m3) * xh(m1)) / (h(m1) - h(m3))
  y2 = (h(m1) * yh(m3) - h(m3) * yh(m1)) / (h(m1) - h(m3))
  GOTO drawit
```

```
case6:
  x1 = xh(m3): y1 = yh(m3)
  x2 = (h(m2) * xh(m1) - h(m1) * xh(m2)) / (h(m2) - h(m1))
  y2 = (h(m2) * yh(m1) - h(m1) * yh(m2)) / (h(m2) - h(m1))
  GOTO drawit
```

```
case7:
  x1 = (h(m2) * xh(m1) - h(m1) * xh(m2)) / (h(m2) - h(m1))
  y1 = (h(m2) * yh(m1) - h(m1) * yh(m2)) / (h(m2) - h(m1))
  x2 = (h(m3) * xh(m2) - h(m2) * xh(m3)) / (h(m3) - h(m2))
  y2 = (h(m3) * yh(m2) - h(m2) * yh(m3)) / (h(m3) - h(m2))
  GOTO drawit
```

```
case8:
  x1 = (h(m3) * xh(m2) - h(m2) * xh(m3)) / (h(m3) - h(m2))
  y1 = (h(m3) * yh(m2) - h(m2) * yh(m3)) / (h(m3) - h(m2))
  x2 = (h(m1) * xh(m3) - h(m3) * xh(m1)) / (h(m1) - h(m3))
  y2 = (h(m1) * yh(m3) - h(m3) * yh(m1)) / (h(m1) - h(m3))
```

```

      GOTO drawit
case9:
  x1 = (h(m1) * xh(m3) - h(m3) * xh(m1)) / (h(m1) - h(m3))
  y1 = (h(m1) * yh(m3) - h(m3) * yh(m1)) / (h(m1) - h(m3))
  x2 = (h(m2) * xh(m1) - h(m1) * xh(m2)) / (h(m2) - h(m1))
  y2 = (h(m2) * yh(m1) - h(m1) * yh(m2)) / (h(m2) - h(m1))
  GOTO drawit
drawit:
  CALL VecOut(x1, y1, x2, y2, K)
case0:
  NEXT M

noneintri:
  NEXT K

noneinbox:
  NEXT I
  NEXT J
RETURN

END SUB

```

```

'$INCLUDE: 'CURRENTB.BI'

REM $DYNAMIC
SUB GlobalHandler
SELECT CASE ERR

'   CASE 5
'       SCREEN 0, 1, 0, 0
'       COLOR 7, 0
'       CLS
'       PRINT "Memory overflow. Reboot system or check hardware configuration"
'       BEEP
'       END

CASE 243
    TextLine$(1) = "You must read in data file first"
    ErrWindow 1, TextLine$(), 243
CASE 244
    TextLine$(1) = "The first point of surface generation limits"
    TextLine$(2) = "is beyond the specified slope"
    ErrWindow 2, TextLine$(), 244
CASE 245
    TextLine$(1) = "The limits of termination of surface generation"
    TextLine$(2) = "exceed slope boundaries in Y direction"
    ErrWindow 2, TextLine$(), 245
CASE 246
    TextLine$(1) = "The upper boundary of the generated sliding surface"
    TextLine$(2) = "can not be above the soil surface"
    TextLine$(3) = "Please, reenter vertical constraints"
    ErrWindow 3, TextLine$(), 246

CASE 247
    TextLine$(1) = "Unexpected program error. Please contact costumer support"
    TextLine$(2) = "Please press a key to continue"
    ErrWindow 2, TextLine$(), 247

CASE 248
    TextLine$(1) = "This file is not a PCSTBL3D critical surface file"
    TextLine$(2) = "Please press any key to continue"
    ErrWindow 2, TextLine$(), 248

CASE 249
    TextLine$(1) = "This file is not a PCSTBL3D data file"
    TextLine$(2) = "Please press any key to continue"
    ErrWindow 2, TextLine$(), 249

CASE 250
    TextLine$(1) = "Valid porepressure model types are 1 or 2 only"
    TextLine$(2) = "Please press a key to continue"
    ErrWindow 2, TextLine$(), 250

CASE 251
    TextLine$(1) = "Maximum number of data points"
    TextLine$(2) = "in Y direction is limited to " + STR$(MAXCOORD)
    ErrWindow 2, TextLine$(), 251

CASE 252
    TextLine$(1) = "Maximum number of data points"
    TextLine$(2) = "in X direction is limited to " + STR$(MAXCOORD)
    ErrWindow 2, TextLine$(), 252

CASE 253
    TextLine$(1) = "Maximum number of soils"
    TextLine$(2) = "is limited to " + STR$(MAXSOIL)
    ErrWindow 2, TextLine$(), 253

CASE 254
    TextLine$(1) = "Input number contains invalid character"
    TextLine$(2) = "Please press a key to continue"

```

```
ErrWindow 2, TextLine$, 254
```

```
CASE 255
```

```
TextLine$(1) = "Invalid file name"
```

```
TextLine$(2) = "Please press a key to continue"
```

```
ErrWindow 2, TextLine$, 255
```

```
CASE ELSE
```

```
TextLine$(1) = "Fatal runtime error. Please contact costumer support"
```

```
TextLine$(2) = "Press a key to continue"
```

```
ErrWindow 2, TextLine$, ERR
```

```
END SELECT
```

```
END SUB
```

```
DECLARE FUNCTION ASCRight% (Arg$)
```

```
'$INCLUDE: 'CURRENTB.BI'
```

```
LocalHandler:
```

```
GlobalHandler
```

```
RESUME NEXT
```

```
REM $DYNAMIC
```

```
DEFINT A-Z
```

```
FUNCTION ASCRight (Arg$)
```

```
'Function AscRight returns ASCII code of the right character of a string
```

```
    ASCRight = ASC(RIGHT$(Arg$, 1))
```

```
END FUNCTION
```

```
DEFSNG A-Z
```

```
SUB CapsCheck (Eflag%) STATIC
```

```
'returns spaces if not on, letters if on
```

```
    DEF SEG = 0
```

```
    CAP$ = "    "
```

```
    NIM$ = "    "
```

```
    IF Eflag% = FALSE THEN IKE$ = "    "
```

```
    IF Eflag% = TRUE THEN IKE$ = "INS"
```

```
    IF PEEK(&H417) AND 64 THEN CAP$ = "CAPS"
```

```
    IF PEEK(&H418) AND 64 THEN CAP$ = "    "
```

```
    IF PEEK(&H417) AND 32 THEN NIM$ = "NUM"
```

```
    IF PEEK(&H418) AND 32 THEN NIM$ = "    "
```

```
    IF (PEEK(&H417) AND 2) OR (PEEK(&H417) AND 1) THEN CAP$ = "CAPS"
```

```
    IF (PEEK(&H418) AND 2) OR (PEEK(&H418) AND 1) THEN CAP$ = "    "
```

```
    CALL FASTPRT(IKE$ + " " + NIM$ + " " + CAP$, 2, 56, 112, RETURN.CODE%)
```

```
END SUB
```

```
DEFINT A-Z
```

```
SUB NewGetIn (FldContent AS STRING, numonly, MaxLen, FldRow, LeftColumn, ColorAttr,  
ReturnVar AS STRING, AbortVar AS STRING) STATIC
```

```
'Author:      G. Moshkovich
```

```
'Created:     9/90
```

```
'Bugs:
```

```
'Purpose:
```

```
'Subroutine NewGetIn creates input field at the specified location, with  
'the specified color and reads string input from this field.
```

```
'It has the same parameters format as the subroutine GetIn, written  
'by J. Thomaz, the difference is the enhanced string editing options and  
'extended variety of the abort variable values.
```

```
'Parameters:
```

```
'  Input:
```

```
'FldContent      Initial content of the input field (shows periods if none)
```

```
'NumOnly         1 if numeric input field otherwise 0
```

```
'MaxLen          Maximum length of the input field
```

```
'FldRow          Row where the input field is located
```

```
'LeftColumn      Left column of the input field
```

```
'ColorAtt        Color attribute of the input field
```

```
'  Output:
```

```
'ReturnVar        Return string
```

```
'AbortVar         Variable returning the ASCII code of the abort character
```

```
'Externals:
```

```
'CapsCheck
```

```
'FastPrt
```



```

    IF FldColumn < LeftColumn + MaxLen - 1 THEN
        IF (FldColumn - LeftColumn + 1) <= LEN(Tmp$) THEN
            FldColumn = FldColumn + 1
            LOCATE FldRow, FldColumn, 1
        END IF
    ELSE
        SOUND 600, 2
    END IF

CASE DEL, BKSP
    IF (Ky = DEL) AND (FldColumn - LeftColumn < LEN(Tmp$)) THEN
        Tmp$ = LEFT$(Tmp$, FldColumn - LeftColumn) + RIGHT$(Tmp$,
LEN(Tmp$) - (FldColumn - LeftColumn) - 1)
    ELSEIF (Ky = BKSP) AND (FldColumn > LeftColumn) THEN
        Tmp$ = LEFT$(Tmp$, FldColumn - LeftColumn - 1) + RIGHT$(Tmp$,
LEN(Tmp$) - (FldColumn - LeftColumn))
        FldColumn = FldColumn - 1
    END IF

    CALL FASTPRT(Tmp$ + SPACE$(MaxLen - LEN(Tmp$)), FldRow, LeftColumn,
ColorAttr, RETURN.CODE%)
    LOCATE FldRow, FldColumn, 1

CASE INS
    IF InsFlag = TRUE THEN
        LOCATE FldRow, FldColumn, 1, 6, 7
        InsFlag = FALSE
    ELSEIF InsFlag = FALSE THEN
        LOCATE FldRow, FldColumn, 1, 0, 7
        InsFlag = TRUE
    END IF

CASE HOME
    FldColumn = LeftColumn
    LOCATE FldRow, FldColumn, 1

CASE ENDK
    IF LEN(Tmp$) < MaxLen THEN
        FldColumn = LeftColumn + LEN(Tmp$)
    ELSE
        FldColumn = LeftColumn + LEN(Tmp$) - 1
    END IF
    LOCATE FldRow, FldColumn, 1

CASE ENTER, F2
    ReturnVar = Tmp$
    EXIT SUB

CASE DOWN, UP, PGDN, PGUP
    AbortVar = CHR$(Ky)
    EXIT SUB

CASE ESC
    AbortVar = CHR$(ESC)
    EXIT SUB

CASE ELSE
    SOUND 600, 2

END SELECT
END IF
LOOP UNTIL Ky = 3
END SUB

DEFSNG A-Z
SUB SoundBeep STATIC
    SOUND 1000, 1: SOUND 1500, 2: SOUND 500, 1
END SUB

```

```
SUB Strip (rax$, clean$) STATIC
  FOR SS = 1 TO LEN(rax$)
    IF MID$(rax$, SS, 2) = " " THEN
      rax$ = LEFT$(rax$, SS - 1)
      EXIT FOR
    END IF
  NEXT SS
  clean$ = rax$
END SUB
```

```

DECLARE FUNCTION LJustStr$ (Arg!)
DECLARE SUB NewGetIn (FldContent AS STRING, numonly%, MaxLen%, FldRow%, LeftColumn%,
ColorAttr%, ReturnVar AS STRING, AbortVar AS STRING)
DECLARE SUB Edit (FldContent AS STRING, numonly%, MaxLen%, FldRow%, LeftColumn%,
ColorAttr%, ReturnVar AS STRING, AbortVar AS STRING)
'$INCLUDE: 'CURRENTB.BI'

```

```
END
```

```

LocalHandler:
GlobalHandler
RESUME NEXT

```

```

REM $DYNAMIC
SUB AskWindow (NLines AS INTEGER, TextLine() AS STRING, KeyPressed AS STRING)

    SCREEN 0, 1, 0, 0
    COLOR 0, 7

    PCOPY 0, 4
    '-----
    'MAKEWIND (ULR%, ULC%, LRR%, LRC%, FRAME%, FORE%, BACK%, GROW%, SHADOW%, LABEL$)
    '-----
    MaxLine$ = TextLine(1)
    FOR I = 1 TO NLines
        IF LEN(TextLine(I)) > LEN(MaxLine$) THEN
            MaxLine$ = TextLine(I)
            MaxInd = I
        END IF
    NEXT

    IF (NLines MOD 2 = 0) THEN
        LROW% = 12 + CINT(NLines / 2)
    ELSE
        LROW% = 11 + CINT(NLines / 2)
    END IF
    UROW% = 13 - CINT(NLines / 2)
    ULC% = 38 - LEN(MaxLine$) / 2
    LRC% = 41 + LEN(MaxLine$) / 2

    CALL MAKEWIND(UROW%, ULC%, LROW%, LRC%, 2, 4, 7, 0, TRUE, "", RETURN.CODE%)

    FOR I = 1 TO NLines
        LOCATE 12 - CINT(NLines / 2) + I, CINT(40 - LEN(TextLine(I)) / 2), 0
        PRINT TextLine(I);
    NEXT I

    KeyPressed = INPUT$(1)
    PCOPY 4, 0

    SCREEN 0, 1, 0, 0

END SUB

```

```
SUB ClearVar
```

```

    '-----reinitialize variables-----
    Nam$ = ""
    nx = 0: ny = 0: ns = 0
    ERASE x, y, z
    xaprin = 0: yamin = 0: dyamin = 0: yamax = 0: dyamax = 0
    ERASE xmin1, dxmin1, xmin2, dxmin2
    ERASE zmin, dzmin
    ERASE Gama, C, Fi
    ERASE utype, ru, uz

END SUB

```

```
SUB ErrWindow (NLines AS INTEGER, TextLine() AS STRING, ErrCode AS INTEGER)
```

```

SCREEN 0, 1, 0, 0
COLOR 0, 7
SoundBeep

PCOPY 0, 4
'-----
'MAKEWIND (ULR%, ULC%, LRR%, LRC%, FRAME%, FORE%, BACK%, GROW%, SHADOW%, LABEL%)
'-----
    MaxLine$ = TextLine(1)
    FOR I = 1 TO NLines
        IF LEN(TextLine(I)) > LEN(MaxLine$) THEN
            MaxLine$ = TextLine(I)
            MaxInd = I
        END IF
    NEXT

    ULC% = 38 - LEN(MaxLine$) / 2
    LRC% = 41 + LEN(MaxLine$) / 2
    IF NLines MOD 2 = 0 THEN
        UROW% = 13 - NLines \ 2
        LROW% = 12 + NLines \ 2
    ELSEIF NLines = 1 THEN
        UROW% = 13 - NLines \ 2
        LROW% = 12 + NLines \ 2 + 1
    ELSE
        UROW% = 13 - NLines \ 2 - 1
        LROW% = 12 + NLines \ 2
    END IF

    CALL MAKEWIND(UROW%, ULC%, LROW%, LRC%, 2, 4, 7, 0, SHADOW, "ERROR " +
STR$(ErrCode), RETURN.CODE%)

    FOR I = 1 TO NLines
        LOCATE 12 - CINT(NLines / 2) + I, CINT(40 - LEN(TextLine(I)) / 2), 0
        PRINT TextLine(I);
    NEXT I

    DO: LOOP WHILE INKEY$ = ""
    PCOPY 4, 0

END SUB

FUNCTION Filter$ (Txt$, FilterString$, Key$) STATIC
    Temp$ = ""
    TxtLength = LEN(Txt$)

    FOR I = 1 TO TxtLength
        C$ = MID$(Txt$, I, 1)
        'Isolate each character in
        'the string

        IF Key$ = "N" THEN
            'If the character is not in the filter string, save it:
            IF INSTR(FilterString$, C$) = 0 THEN
                Temp$ = Temp$ + C$
            END IF
        ELSEIF Key$ = "Y" THEN
            'If the character is in the filter string, save it:
            IF INSTR(FilterString$, C$) <> 0 THEN
                Temp$ = Temp$ + C$
            END IF
        END IF
    NEXT I

    Filter$ = Temp$
END FUNCTION

SUB InputData (ExitCode AS INTEGER)

    ON ERROR GOTO LocalHandler

```

KEY(15) ON
KEY(16) ON

ExitCode = 0
Footer

'Write a footer

'-----
'-----Start main input loop-----
'-----

DO

SELECT CASE ExitCode

CASE 0, 1

'-----
'Open data file
OpenDataFile ExitCode
IF ExitCode = 666 THEN
    EXIT DO
ELSE
    ExitCode = 10
END IF

CASE 10, 11

'-----
' Input number of points in x and y direction, and number of soils
NumOfGridPoints ExitCode
IF ExitCode = 666 THEN
    ExitCode = 1
ELSE
    ExitCode = 20
END IF

CASE 20, 22

'-----
' Input coordinates X and Y of points
XYCoord ExitCode
IF ExitCode = 666 THEN
    ExitCode = 11
ELSE
    ExitCode = 30
END IF

CASE 30, 33

'-----
' Input coordinates Z of the soil layers for each XY pair
SoilCoord ExitCode
IF ExitCode = 666 THEN
    ExitCode = 22
ELSE
    ExitCode = 40
END IF

CASE 40, 44

'-----
'Input surface generation data: main axis number N, and
'initiation and termination limits along the main axis
SurfGenerData1 ExitCode
IF ExitCode = 666 THEN
    ExitCode = 33
ELSE
    ExitCode = 50
END IF

CASE 50, 55

'-----
'Input surface generation data: left and right initiation values
'on the axis parallel to main
SurfGenerData2 ExitCode
IF ExitCode = 666 THEN

```

        ExitCode = 44
    ELSE
        ExitCode = 60
    END IF

CASE 60, 66
    '-----
    'Input searching constraints in vertical direction
    VerticConstr ExitCode
    IF ExitCode = 666 THEN
        ExitCode = 55
    ELSE
        ExitCode = 70
    END IF

CASE 70, 77
    '-----
    'Input Soil properties data: unit weight, cohesion and shear
    'angle for each soil.
    SoilPropData ExitCode
    IF ExitCode = 666 THEN
        ExitCode = 66
    ELSE
        ExitCode = 80
    END IF

CASE 80, 88
    '-----
    'Input pore pressure data
    PorePresData ExitCode
    IF ExitCode = 666 THEN
        ExitCode = 77
    ELSE
        ExitCode = 90
    END IF

CASE 90
    '-----
    'Print data to the file
    PrintData ExitCode
    IF ExitCode = 666 THEN
        ExitCode = 88
    ELSE
        ExitCode = 100
    END IF

CASE ELSE
    ExitCode = 100
END SELECT
LOOP UNTIL ExitCode = 100

IF ExitCode <> 666 THEN

    TextLine$(1) = "Would you like to check geometry"
    TextLine$(2) = "of the data file you just edited ? y/[n]"
    AskWindow 2, TextLine$, KeyPressed$
    IF KeyPressed$ = "y" OR KeyPressed$ = "Y" THEN
        Caller$ = "geometry"
        Contour
    END IF
END IF
ClearScrn
KEY(5) OFF

```

END SUB

FUNCTION LJustStr\$ (Arg) STATIC
 'This function converts numeric argument into the left justified string


```

LJustStr$ = LTRIM$(STR$(Arg))

END FUNCTION

SUB NotifyWindow (Line1 AS STRING, Line2 AS STRING)
  SCREEN 0, 1, 0, 0
  COLOR 0, 7

  '-----
  'MAKEWIND (ULR%, ULC%, LRR%, LRC%, FRAME%, FORE%, BACK%, GROW%, SHADOW%, LABEL$)
  '-----
  IF LEN(Line1) > LEN(Line2) THEN
    ULC% = 38 - LEN(Line1) / 2
    LRC% = 41 + LEN(Line1) / 2
  ELSE
    ULC% = 38 - LEN(Line2) / 2
    LRC% = 41 + LEN(Line2) / 2
  END IF

  CALL MAKEWIND(10, ULC%, 11, LRC%, 2, 4, 7, 0, TRUE, "", RETURN.CODE%)
  COLOR 16, 7
  LOCATE 10, 40 - LEN(Line1) / 2, 0: PRINT Line1
  LOCATE 11, 40 - LEN(Line2) / 2, 0: PRINT Line2;
  DO: LOOP WHILE INKEY$ = ""
  COLOR 0, 7

END SUB

SUB NumOfGridPoints (ExitCode AS INTEGER)

  SHARED nx, ny, ns

  'This subroutine inputs number of points in X and Y direction and
  'number of soil layers

  PCOPY 0, 1

  CALL MAKEWIND(5, LCOL - 2, 12, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

  CALL FASTPRT("You will be entering the grid of X and Y points next.", 5, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("This grid will be used to represent the soil profile.", 6, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("                                ", 7, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Please enter:  Number of points in X direction:      ", 8, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("                                Number of points in Y direction:      ", 9, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("                                Maximum number of Soils:          ", 10, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("                                ", 11, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("(Remember that the slide occurs in the Y direction) ", 12, LCOL,
49, RETURN.CODE%)

  CALL FASTPRT(LJustStr$(nx) + SPACE$(3 - LEN(LJustStr$(nx))), 8, 60, 10,
RETURN.CODE%)
  CALL FASTPRT(LJustStr$(ny) + SPACE$(3 - LEN(LJustStr$(ny))), 9, 60, 10,
RETURN.CODE%)
  CALL FASTPRT(LJustStr$(ns) + SPACE$(3 - LEN(LJustStr$(ns))), 10, 53, 10,
RETURN.CODE%)

  DO
    SELECT CASE ExitCode

      CASE 10
        'Input number of points in X direction

```

```

NewGetIn LJustStr$(nx), 1, 3, 8, 60, 10, nx$, ab$
IF ab$ = CHR$(ESC) THEN
    ExitCode = 666          '666-"Devils" exitcode
    PCOPY 1, 0
    EXIT SUB
ELSEIF VAL(nx$) > MAXCOORD THEN
    ERROR 252
ELSE
    nx = VAL(nx$)
    ExitCode = 900
END IF

CASE 900
    'Input number of points in Y direction-----
NewGetIn LJustStr$(ny), 1, 3, 9, 60, 10, ny$, ab$
IF ab$ = CHR$(ESC) THEN
    CALL FASTPRT(LJustStr$(ny) + SPACE$(1), 9, 60, 10, RETURN.CODE%)
    ExitCode = 10
ELSEIF VAL(ny$) > MAXCOORD THEN
    ERROR 251
ELSE
    ny = VAL(ny$)
    ExitCode = 11
END IF

CASE 11
    'Input number of soils -----
NewGetIn LJustStr$(ns), 1, 3, 10, 53, 10, ns$, ab$
IF ab$ = CHR$(ESC) THEN
    CALL FASTPRT(LJustStr$(ns) + SPACE$(1), 10, 53, 10, RETURN.CODE%)
    ExitCode = 900
ELSEIF (Filter(ns$, "1234567890.", "N") <> "") OR (VAL(ns$) = 0) THEN
    ERROR 254
    ns$ = ""
ELSEIF VAL(ns$) > MAXSOIL THEN
    ERROR 253
ELSE
    ns = VAL(ns$)
    ExitCode = 901
END IF

CASE ELSE
    'Just in case.-----
    ERROR 247
    ExitCode = 666
    PCOPY 1, 0
    EXIT SUB

END SELECT
LOOP UNTIL ExitCode = 901
PCOPY 1, 0
END SUB

SUB OpenDataFile (ExitCode AS INTEGER)

    '-----
    'This subroutine opens data file and returns ExitCode=666 if ESC was
    'pressed or returns ExitCode=0 as a result of succesfull finishing
    '-----

    CALL MAKEWIND(2, 3, 18, 78, 2, 11, 1, 0, FALSE, "", RETURN.CODE%)
    ShowMes$ = "Press CNTRL-END to suspend edition"
    CALL FASTPRT(ShowMes$, 18, 22, 49, RETURN.CODE%)

    ' Input filename

    PCOPY 0, 1
    CALL MAKEWIND(5, LCOL - 2, 9, RCOL + 2, 2, 1, 11, GROW, SHADOW, "", RETURN.CODE%)

```

```

CALL FASTPRT("Enter a Name for your Data File:           ", 5,
LCOL, 49, RETURN.CODE%)
CALL FASTPRT("                                           ", 6,
LCOL, 49, RETURN.CODE%)
CALL FASTPRT("Your data will be written to this file.           ", 7,
LCOL, 49, RETURN.CODE%)
CALL FASTPRT("If a File with the same name already exists, it will be  ", 8,
LCOL, 49, RETURN.CODE%)
CALL FASTPRT("overwritten, so please make sure you use an original name!", 9,
LCOL, 49, RETURN.CODE%)

DO
  NewGetIn Nam$, 0, 12, 5, 45, 10, TMPNam$, ab$

  IF ab$ = CHR$(ESC) THEN

    TextLine$(1) = "Your changes will be lost if you exit edit mode"
    TextLine$(2) = "Please press 'y' to confirm this operation"
    TextLine$(3) = "Press any other key to continue editing"
    AskWindow 3, TextLine$(), KeyPressed$

    IF UCASE$(KeyPressed$) = "Y" THEN
      ExitCode = 666
      EXIT SUB
    END IF

    ELSEIF Filter$(TMPNam$, " ", "N") = "" THEN
      ERROR 255
      ExitCode = 667
      TMPNam$ = ""
    ELSE
      ExitCode = 999
      Nam$ = UCASE$(Filter$(TMPNam$, " ", "N"))
    END IF
  LOOP UNTIL ExitCode = 999

  PCOPY 1, 0

END SUB

SUB PiezoLine (ExitCode AS INTEGER, K)
  ExitCode = 903
  I = 1                                'initilize variables
  J = 1

  PCOPY 0, 1
  CALL MAKEWIND(5, LCOL - 2, 14, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

  CALL FASTPRT("PORE PRESSURE DATA - Piezometric Line Data           ", 5, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("For soil number      Please enter the piezometric value", 7, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("for the coordinates below:                             ", 8, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("X(      ) =                                           ", 10, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Y(      ) =                                           ", 11, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("====> piezometric value =                                           ", 13, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("          !!! REMEMBER TO USE COMPATIBLE UNITS !!!           ", 14, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT(STR$(K), 7, 27, 59, RETURN.CODE%)

DO                                     'begin I loop
  DO                                   'begin J loop
    SELECT CASE ExitCode

```

CASE 903

```
CALL FASTPRT(STR$(I) + SPACE$(1), 10, 14, 59, RETURN.CODE%)
CALL FASTPRT(STR$(x(I)) + SPACE$(4), 10, 21, 59, RETURN.CODE%)
CALL FASTPRT(STR$(J) + SPACE$(1), 11, 14, 59, RETURN.CODE%)
CALL FASTPRT(STR$(y(J)) + SPACE$(4), 11, 21, 59, RETURN.CODE%)
NewGetIn LJustStr$(uz(I, J, K)), 1, 7, 13, 38, 10, uz$, ab$
```

```
IF ab$ = CHR$(ESC) AND I = 1 AND J = 1 THEN
  ExitCode = 666
  PCOPY 1, 0
  EXIT SUB
ELSEIF ab$ = CHR$(ESC) AND J = 1 THEN
  I = I - 1
  J = ny
  ExitCode = 903
ELSEIF ab$ = CHR$(ESC) THEN
  J = J - 1
  ExitCode = 903
ELSEIF Filter(uz$, "-1234567890.", "N") <> "" THEN
  ERROR 254
  uz$ = ""
ELSE
  uz(I, J, K) = VAL(uz$)
  'PRINT #3, uz(I, J, K);
  ExitCode = 903
  J = J + 1
END IF
```

```
END SELECT
LOOP WHILE J <= ny
```

```
'PRINT #3,
J = 1
I = I + 1
LOOP WHILE I <= nx
PCOPY 1, 0
END SUB
```

SUB PorePresData (ExitCode AS INTEGER)

```
IF ExitCode = 80 THEN
  K = 1
ELSEIF ExitCode = 88 THEN
  K = ns
  ExitCode = 80
END IF
```

```
DO
  'begin K loop
```

```
  SELECT CASE ExitCode
    CASE 80
```

```
    PCOPY 0, 1
    CALL MAKEWIND(5, LCOL - 2, 15, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)
```

```
    CALL FASTPRT("PORE PRESSURE DATA:
", 5, LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("Please select the porepressure model type from the
list", 7, LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("below:
", 8, LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("1 - Pore pressure ratio (Ru)
", 10, LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("2 - Piezometric Surfaces
", 11, LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("For soil number      , the model you choose is:
", 13, LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("
", 14, LCOL, 49, RETURN.CODE%)
    (Enter 1 or 2)
```

```

ExitCode = 900

CASE 900
CALL FASTPRT(STR$(K), 13, 27, 59, RETURN.CODE%)
NewGetIn LJustStr$(utype(K)), 1, 1, 13, 58, 10, utype$, ab$
IF ab$ = CHR$(ESC) AND K = 1 THEN
ExitCode = 666
PCOPY 1, 0
EXIT SUB
ELSEIF ab$ = CHR$(ESC) THEN
ExitCode = 900
K = K - 1
ELSEIF Filter(utype$, "-1234567890.", "N") <> "" THEN
ERROR 254
utype$ = ""
ELSE
utype(K) = VAL(utype$)
'PRINT #3, utype(K);
'PRINT #3,

IF utype(K) = 1 THEN
PCOPY 0, 2
CALL FASTPRT("Then enter Ru for this soil: ", 15, LCOL, 49,
RETURN.CODE%)

NewGetIn LJustStr$(ru(K)), 1, 6, 15, 43, 10, ru$, ab$
IF ab$ = CHR$(ESC) THEN
PCOPY 2, 0
ExitCode = 900
ELSEIF Filter(ru$, "-1234567890.", "N") <> "" THEN
ERROR 254
ru$ = ""
ELSE
ru(K) = VAL(ru$)
'PRINT #3, ru
ExitCode = 900
K = K + 1
PCOPY 2, 0
END IF

ELSEIF utype(K) = 2 THEN
PCOPY 1, 0
PiezoLine ExitCode, K
IF ExitCode = 666 THEN
ExitCode = 80
ELSE
ExitCode = 80
K = K + 1
END IF
ELSE
ERROR 250
END IF
END IF
END SELECT
LOOP WHILE K <= ns
PCOPY 1, 0
END SUB

SUB PrintData (ExitCode AS INTEGER)

TextLine$(1) = "Your data are about to be written to the file " + UCASE$(Nam$)
TextLine$(2) = "To confirm operation press 'y'"
TextLine$(3) = "Any other key will bring you back"

AskWindow 3, TextLine$(), KeyPressed$

IF UCASE$(KeyPressed$) = "Y" THEN
PCOPY 0, 4
NotifyWindow "Please wait", "Your data are been procesed"

```

```

ELSE
    ExitCode = 666
    EXIT SUB
END IF

OPEN Nam$ FOR OUTPUT AS #3

'----- file header for data file -----
PRINT #3, "PCSTBL3D Version 2.0"

'----- print number of data points and number of soils ---
PRINT #3, nx, ny
PRINT #3, ns
'----- print x coordinates -----
FOR I = 1 TO nx
    PRINT #3, x(I);
NEXT I
PRINT #3,
'----- print y coordinates -----
FOR I = 1 TO ny
    PRINT #3, y(I);
NEXT I
PRINT #3,
'----- print soil coordinates -----
FOR K = 1 TO ns
    FOR I = 1 TO nx
        FOR J = 1 TO ny
            PRINT #3, z(I, J, K);
        NEXT J
    PRINT #3,
    NEXT I
NEXT K
'----- print surface generation data 1 -----
PRINT #3, xaprin
PRINT #3, yamin; dyamin; yamax; dyamax
'----- print surface generation data 2 -----
FOR I = 1 TO ny
    PRINT #3, xmin1(I); dxmin1(I); xmin2(I); dxmin2(I)
NEXT I
'----- print vertical constarints -----
FOR I = 1 TO nx
    FOR J = 1 TO ny
        PRINT #3, zmin(I, J); dzmin(I, J);
    NEXT J
    PRINT #3,
NEXT I
'----- print soil properties -----
FOR I = 1 TO ns
    PRINT #3, Gama(I), C(I), Fi(I)
NEXT I
'----- print pore pressure data -----
FOR K = 1 TO ns
    PRINT #3, utype(K);
    PRINT #3,
    IF utype(K) = 1 THEN
        PRINT #3, ru(K)
    ELSE
        FOR I = 1 TO nx
            FOR J = 1 TO ny
                PRINT #3, uz(I, J, K);
            NEXT J
        PRINT #3,
        NEXT I
    END IF
NEXT K
CLOSE
PCOPY 4, 0

```

END SUB

```
SUB SoilCoord (ExitCode AS INTEGER)
```

```

  IF ExitCode = 30 THEN
    IndexM = 1 'Initialize index variables
    IndexI = 1
    IndexJ = 1
  ELSEIF ExitCode = 33 THEN
    IndexM = ns
    IndexI = nx
    IndexJ = ny
  END IF

  PCOPY 0, 1
  CALL MAKEWIND(5, LCOL - 2, 11, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

  CALL FASTPRT("Now you will enter the soil layers coordinates for      ", 5, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("each grid point:                                     ", 6, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("For SOIL number:                                     ", 8, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("For point X(      ), and Y(      ) enter value of Z =   ", 9, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Remember to use consistent length units...           ", 11, LCOL,
49, RETURN.CODE%)

  FOR M = IndexM TO ns
    FOR I = IndexI TO nx
      FOR J = IndexJ TO ny

        CALL FASTPRT(STR$(M), 8, 28, 59, RETURN.CODE%)
        CALL FASTPRT(STR$(I) + SPACE$(1), 9, 24, 59, RETURN.CODE%)
        CALL FASTPRT(STR$(J) + SPACE$(1), 9, 37, 59, RETURN.CODE%)

        NewGetIn LJustStr$(z(I, J, M)), 1, 6, 9, 63, 10, z$, ab$

        IF ab$ = CHR$(ESC) AND M = 1 AND I = 1 AND J = 1 THEN
          ExitCode = 666
          PCOPY 1, 0
          EXIT SUB
        ELSEIF ab$ = CHR$(ESC) AND I = 1 AND J = 1 THEN
          M = M - 1
          I = ny
          J = ny - 1
        ELSEIF ab$ = CHR$(ESC) AND J = 1 THEN
          I = I - 1
          J = ny - 1
        ELSEIF ab$ = CHR$(ESC) THEN
          J = J - 2
        ELSEIF Filter(z$, "-1234567890.", "N") <> "" THEN
          ERROR 254
          z$ = ""
          J = J - 1
        ELSE
          z(I, J, M) = VAL(z$)
          'PRINT #3, z(I, J, M);
        END IF
      NEXT J
    NEXT I
  NEXT M
  PCOPY 1, 0
END SUB

```

```
SUB SoilPropData (ExitCode AS INTEGER)
```

```

  IF ExitCode = 70 THEN
    I = 1

```



```

ELSE
    I = ns
END IF

PCOPY 0, 1
CALL MAKEWIND(5, LCOL - 2, 16, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

CALL FASTPRT("Soil Properties Data:                                ", 5, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("Now you will enter the strength properties of", 7, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("the soils:                                          ", 8, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("For soil number      ,enter:                        ", 10, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("Unit weight .....                                ", 12, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("Cohesion .....                                    ", 13, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("Shear Angle .....                                ", 14, LCOL, 49,
RETURN.CODE%)
CALL FASTPRT("    !!! REMEMBER TO USE COMPATIBLE UNITS !!! ", 16, LCOL, 49,
RETURN.CODE%)

CALL FASTPRT(LJustStr$(Gama(I)) + SPACE$(6 - LEN(LJustStr$(Gama(I)))), 12, 36,
10, RETURN.CODE%)
CALL FASTPRT(LJustStr$(C(I)) + SPACE$(6 - LEN(LJustStr$(C(I)))), 13, 36, 10,
RETURN.CODE%)
CALL FASTPRT(LJustStr$(Fi(I)) + SPACE$(6 - LEN(LJustStr$(Fi(I)))), 14, 36, 10,
RETURN.CODE%)

DO
    SELECT CASE ExitCode

        CASE 70
            CALL FASTPRT(STR$(I), 10, 28, 59, RETURN.CODE%)
            NewGetIn LJustStr$(Gama(I)), 1, 6, 12, 36, 10, Gama$, ab$
            IF ab$ = CHR$(ESC) AND I = 1 THEN
                ExitCode = 666
                PCOPY 1, 0
                EXIT SUB
            ELSEIF ab$ = CHR$(ESC) THEN
                I = I - 1
                CALL FASTPRT(LJustStr$(Gama(I)) + SPACE$(6 -
LEN(LJustStr$(Gama(I)))), 12, 36, 10, RETURN.CODE%)
                CALL FASTPRT(LJustStr$(C(I)) + SPACE$(6 - LEN(LJustStr$(C(I)))), 13,
36, 10, RETURN.CODE%)
                CALL FASTPRT(LJustStr$(Fi(I)) + SPACE$(6 - LEN(LJustStr$(Fi(I)))),
14, 36, 10, RETURN.CODE%)
                ExitCode = 77
            ELSEIF Filter(Gama$, "-1234567890.", "N") <> "" THEN
                ERROR 254
                Gama$ = ""
            ELSE
                Gama(I) = VAL(Gama$)
                ExitCode = 900
            END IF

        CASE 900
            NewGetIn LJustStr$(C(I)), 1, 6, 13, 36, 10, C$, ab$
            IF ab$ = CHR$(ESC) THEN
                ExitCode = 70
            ELSEIF Filter(C$, "-1234567890.", "N") <> "" THEN
                ERROR 254
                C$ = ""
            ELSE
                C(I) = VAL(C$)
                ExitCode = 77
            END IF
    
```

```

END IF

CASE 77
  CALL FASTPRT(STR$(I), 10, 28, 59, RETURN.CODE%)
  NewGetIn LJustStr$(Fi(I)), 1, 6, 14, 36, 10, Fi$, ab$
  IF ab$ = CHR$(ESC) THEN
    ExitCode = 900
  ELSEIF Filter(Fi$, "-1234567890.", "N") <> "" THEN
    ERROR 254
    Fi$ = ""
  ELSE
    Fi(I) = VAL(Fi$)
    'PRINT #3, Gama, C, Fi
    ExitCode = 70
    I = I + 1
    CALL FASTPRT(LJustStr$(Gama(I)) + SPACE$(6 -
LEN(LJustStr$(Gama(I)))), 12, 36, 10, RETURN.CODE%)
    CALL FASTPRT(LJustStr$(C(I)) + SPACE$(6 - LEN(LJustStr$(C(I)))), 13,
36, 10, RETURN.CODE%)
    CALL FASTPRT(LJustStr$(Fi(I)) + SPACE$(6 - LEN(LJustStr$(Fi(I)))),
14, 36, 10, RETURN.CODE%)

    END IF

  END SELECT

  LOOP WHILE I <= ns
  PCOPY 1, 0
END SUB

SUB SurfGenerDatal (ExitCode AS INTEGER)

  PCOPY 0, 1
  CALL MAKEWIND(5, LCOL - 2, 15, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

  CALL FASTPRT("Data for Surface Generation:                                ", 5, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Enter number N where X(N) would be the Principal Axis ", 7, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("of Sliding (ex: x(3)-> N=3) .....                                ", 8, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("On the principal axis of Sliding, enter search limits:", 10, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Initiation value of Y,(Ymin) .....                                ", 12, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Range of Ymin .....                                ", 13, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Termination value of Y, (Ymax) .....                                ", 14, LCOL,
49, RETURN.CODE%)
  CALL FASTPRT("Range of Ymax .....                                ", 15, LCOL,
49, RETURN.CODE%)

  CALL FASTPRT(LJustStr$(xaprin) + SPACE$(3 - LEN(LJustStr$(xaprin))), 8, 60, 10,
RETURN.CODE%)
  CALL FASTPRT(LJustStr$(yamin) + SPACE$(3 - LEN(LJustStr$(yamin))), 12, 60, 10,
RETURN.CODE%)
  CALL FASTPRT(LJustStr$(dyamin) + SPACE$(3 - LEN(LJustStr$(dyamin))), 13, 60, 10,
RETURN.CODE%)
  CALL FASTPRT(LJustStr$(yamax) + SPACE$(3 - LEN(LJustStr$(yamax))), 14, 60, 10,
RETURN.CODE%)
  CALL FASTPRT(LJustStr$(dyamax) + SPACE$(3 - LEN(LJustStr$(dyamax))), 15, 60, 10,
RETURN.CODE%)

DO
  SELECT CASE ExitCode

    CASE 40
      NewGetIn LJustStr$(xaprin), 1, 3, 8, 60, 10, xaprin$, ab$
      IF ab$ = CHR$(ESC) THEN

```

```

        ExitCode = 666
        PCOPY 1, 0
        EXIT SUB
    ELSEIF Filter(xaprin$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        xaprin$ = ""
    ELSE
        xaprin = VAL(xaprin$)
        ExitCode = 900
    END IF

CASE 900
    NewGetIn LJustStr$(yamin), 1, 3, 12, 60, 10, yamin$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(yamin), 12, 60, 10, RETURN.CODE%)
        ExitCode = 40
    ELSEIF Filter(yamin$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        yamin$ = ""
    ELSE
        yamin = VAL(yamin$)
        IF yamin <= y(1) THEN
            ERROR 244
            ExitCode = 900
        ELSE
            ExitCode = 901
        END IF
    END IF

CASE 901
    NewGetIn LJustStr$(dyamin), 1, 3, 13, 60, 10, dyamin$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(dyamin), 13, 60, 10, RETURN.CODE%)
        ExitCode = 900
    ELSEIF Filter(dyamin$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        dyamin$ = ""
    ELSE
        dyamin = VAL(dyamin$)
        ExitCode = 902
    END IF

CASE 902
    NewGetIn LJustStr$(yamax), 1, 3, 14, 60, 10, yamax$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(yamax), 14, 60, 10, RETURN.CODE%)
        ExitCode = 901
    ELSEIF Filter(yamax$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        yamax$ = ""
    ELSE
        ExitCode = 44
        yamax = VAL(yamax$)
    END IF

CASE 44
    NewGetIn LJustStr$(dyamax), 1, 3, 15, 60, 10, dyamax$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(dyamax), 15, 60, 10, RETURN.CODE%)
        ExitCode = 902
    ELSEIF Filter(dyamax$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        dyamax$ = ""
    ELSE
        dyamax = VAL(dyamax$)
        IF yamax + dyamax >= y(ny) THEN
            ERROR 245
            ExitCode = 44
        ELSE

```

```

        ExitCode = 999
    END IF
END IF

END SELECT

LOOP UNTIL ExitCode = 999
PCOPY 1, 0
END SUB

SUB SurfGenerData2 (ExitCode AS INTEGER)

    IF ExitCode = 50 THEN
        IndexI = 1
    ELSE
        IndexI = ny
    END IF

    PCOPY 0, 1
    CALL MAKEWIND(5, LCOL - 2, 15, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

    CALL FASTPRT("Data for Surface Generation:                                ", 5,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("You will enter boundary conditions for the search                ", 7,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("specifying the width of the random surfaces:                    ", 8,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("On the axis parallel to X on the coordinate Y(      ), enter:", 10,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("Left initiation value of X .....                            ", 12,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("Range of left X .....                                    ", 13,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("Right initiation value of X .....                            ", 14,
LCOL, 49, RETURN.CODE%)
    CALL FASTPRT("Range of right X.....                                    ", 15,
LCOL, 49, RETURN.CODE%)

    FOR I = IndexI TO ny

        CALL FASTPRT(STR$(I) + SPACE$(1), 10, 58, 59, RETURN.CODE%)
        CALL FASTPRT(LJustStr$(xmin1(I)) + SPACE$(3 - LEN(LJustStr$(xmin1(I)))), 12,
59, 10, RETURN.CODE%)
        CALL FASTPRT(LJustStr$(dxmin1(I)) + SPACE$(3 - LEN(LJustStr$(dxmin1(I)))), 13,
59, 10, RETURN.CODE%)
        CALL FASTPRT(LJustStr$(xmin2(I)) + SPACE$(3 - LEN(LJustStr$(xmin2(I)))), 14,
59, 10, RETURN.CODE%)
        CALL FASTPRT(LJustStr$(dxmin2(I)) + SPACE$(3 - LEN(LJustStr$(dxmin2(I)))), 15,
59, 10, RETURN.CODE%)

    DO
        SELECT CASE ExitCode

            CASE 50
                NewGetIn LJustStr$(xmin1(I)), 1, 3, 12, 59, 10, xmin1$, ab$
                IF ab$ = CHR$(ESC) AND I = 1 THEN
                    ExitCode = 666
                    PCOPY 1, 0
                    EXIT SUB
                ELSEIF ab$ = CHR$(ESC) THEN
                    I = I - 1
                    CALL FASTPRT(STR$(I) + SPACE$(1), 10, 58, 59, RETURN.CODE%)
                    CALL FASTPRT(LJustStr$(xmin1(I)) + SPACE$(3 -
LEN(LJustStr$(xmin1(I)))), 12, 59, 10, RETURN.CODE%)
                    CALL FASTPRT(LJustStr$(dxmin1(I)) + SPACE$(3 -
LEN(LJustStr$(dxmin1(I)))), 13, 59, 10, RETURN.CODE%)

```

```

        CALL FASTPRT(LJustStr$(xmin2(I)) + SPACE$(3 -
LEN(LJustStr$(xmin2(I)))), 14, 59, 10, RETURN.CODE%)
        CALL FASTPRT(LJustStr$(dxmin2(I)) + SPACE$(3 -
LEN(LJustStr$(dxmin2(I)))), 15, 59, 10, RETURN.CODE%)
        ExitCode = 55
    ELSEIF Filter(xmin1$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        xmin1$ = ""
    ELSE
        xmin1(I) = VAL(xmin1$)
        ExitCode = 900
    END IF

CASE 900
    NewGetIn LJustStr$(dxmin1(I)), 1, 3, 13, 59, 10, dxmin1$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(dxmin1(I)), 13, 59, 10, RETURN.CODE%)
        ExitCode = 50
    ELSEIF Filter(dxmin1$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        dxmin1$ = ""
    ELSE
        dxmin1(I) = VAL(dxmin1$)
        ExitCode = 901
    END IF

CASE 901
    NewGetIn LJustStr$(xmin2(I)), 1, 3, 14, 59, 10, xmin2$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(xmin2(I)), 14, 59, 10, RETURN.CODE%)
        ExitCode = 900
    ELSEIF Filter(xmin2$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        xmin2$ = ""
    ELSE
        xmin2(I) = VAL(xmin2$)
        ExitCode = 55
    END IF

CASE 55
    NewGetIn LJustStr$(dxmin2(I)), 1, 3, 15, 59, 10, dxmin2$, ab$
    IF ab$ = CHR$(ESC) THEN
        CALL FASTPRT(LJustStr$(dxmin2(I)), 15, 59, 10, RETURN.CODE%)
        ExitCode = 901
    ELSEIF Filter(dxmin2$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        dxmin2$ = ""
    ELSE
        ExitCode = 999
        dxmin2(I) = VAL(dxmin2$)
    END IF

END SELECT

LOOP UNTIL ExitCode = 999

IF I < ny THEN
    ExitCode = 50
END IF

NEXT I
PCOPY 1, 0

END SUB

SUB VerticConstr (ExitCode AS INTEGER)

IF ExitCode = 60 THEN
    I = 1

```

```

        J = 1
    ELSE
        I = nx
        J = ny
    END IF

    PCOPY 0, 1
    CALL MAKEWIND(5, LCOL - 2, 11, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

    CALL FASTPRT("Now you will enter the searching constrains for the ", 5, LCOL,
49, RETURN.CODE%)
    CALL FASTPRT("search in the Z (vertical) direction for all the grid:", 6, LCOL,
49, RETURN.CODE%)
    CALL FASTPRT("For point X( ), and Y( ) enter value of Zmin = ", 9, LCOL,
49, RETURN.CODE%)
    CALL FASTPRT("                                delta Z = ", 10, LCOL,
49, RETURN.CODE%)
    CALL FASTPRT("Remember to use consistent length units... ", 11, LCOL,
49, RETURN.CODE%)

    CALL FASTPRT(LJustStr$(zmin(I, J)) + SPACE$(6 - LEN(LJustStr$(zmin(I, J)))), 9,
65, 10, RETURN.CODE%)
    CALL FASTPRT(LJustStr$(dzmin(I, J)) + SPACE$(6 - LEN(LJustStr$(dzmin(I, J)))),
10, 65, 10, RETURN.CODE%)

    DO                                'start I loop
    DO                                'start J loop
        SELECT CASE ExitCode

            CASE 60
                CALL FASTPRT(STR$(I) + SPACE$(1), 9, 24, 59, RETURN.CODE%)
                CALL FASTPRT(STR$(J) + SPACE$(1), 9, 37, 59, RETURN.CODE%)
                CALL FASTPRT(LJustStr$(zmin(I, J)) + SPACE$(6 - LEN(LJustStr$(zmin(I,
J)))), 9, 65, 10, RETURN.CODE%)
                CALL FASTPRT(LJustStr$(dzmin(I, J)) + SPACE$(6 -
LEN(LJustStr$(dzmin(I, J)))), 10, 65, 10, RETURN.CODE%)
                NewGetIn LJustStr$(zmin(I, J)), 1, 6, 9, 65, 10, zmin$, ab$
                IF ab$ = CHR$(ESC) AND I = 1 AND J = 1 THEN
                    ExitCode = 666
                    PCOPY 1, 0
                    EXIT SUB
                ELSEIF ab$ = CHR$(ESC) AND J = 1 THEN
                    I = I - 1
                    J = ny
                    CALL FASTPRT(LJustStr$(zmin(I, J)), 9, 65, 10, RETURN.CODE%)
                    ExitCode = 66
                ELSEIF ab$ = CHR$(ESC) THEN
                    J = J - 1
                    CALL FASTPRT(LJustStr$(zmin(I, J)), 9, 65, 10, RETURN.CODE%)
                    ExitCode = 66
                ELSEIF Filter(zmin$, "-1234567890.", "N") <> "" THEN
                    ERROR 254
                    zmin$ = ""
                ELSE
                    zmin(I, J) = VAL(zmin$)
                    ExitCode = 66
                END IF

            CASE 66
                CALL FASTPRT(STR$(I) + SPACE$(1), 9, 24, 59, RETURN.CODE%)
                CALL FASTPRT(STR$(J) + SPACE$(1), 9, 37, 59, RETURN.CODE%)
                CALL FASTPRT(LJustStr$(zmin(I, J)) + SPACE$(6 - LEN(LJustStr$(zmin(I,
J)))), 9, 65, 10, RETURN.CODE%)
                CALL FASTPRT(LJustStr$(dzmin(I, J)) + SPACE$(6 -
LEN(LJustStr$(dzmin(I, J)))), 10, 65, 10, RETURN.CODE%)
                NewGetIn LJustStr$(dzmin(I, J)), 1, 6, 10, 65, 10, dzmin$, ab$
                IF ab$ = CHR$(ESC) THEN
                    CALL FASTPRT(LJustStr$(dzmin(I, J)), 10, 65, 10, RETURN.CODE%)

```



```

        ExitCode = 60
    ELSEIF Filter(dzmin$, "-1234567890.", "N") <> "" THEN
        ERROR 254
        dzmin$ = ""
    ELSE
        dzmin(I, J) = VAL(dzmin$)
        IF zmin(I, J) + dzmin(I, J) >= z(I, J, 1) THEN
            ERROR 246
        ELSE
            J = J + 1
            ExitCode = 60
        END IF
    END IF

END SELECT

LOOP WHILE J <= ny
'PRINT #3,

I = I + 1
J = 1
LOOP WHILE I <= nx
PCOPY 1, 0

END SUB

SUB XYCoord (ExitCode AS INTEGER)

    SHARED x(), y()
    IF ExitCode = 20 THEN
        IndexX = 1      'Initialize variables
        IndexY = 1
    ELSEIF ExitCode = 22 THEN
        IndexX = nx
        IndexY = ny
    ELSE
        END IF

    PCOPY 0, 1
    CALL MAKEWIND(5, LCOL - 2, 11, RCOL + 2, 2, 1, 11, GROW, SHADOW, "",
RETURN.CODE%)

    CALL FASTPRT("Now you will enter the actual coordinates of the ", 5, LCOL, 49,
RETURN.CODE%)
    CALL FASTPRT("points... ", 6, LCOL, 49,
RETURN.CODE%)
    CALL FASTPRT(" ", 7, LCOL, 49,
RETURN.CODE%)
    CALL FASTPRT("For point X( ), enter its coordinate value: x = ", 8, LCOL, 49,
RETURN.CODE%)
    CALL FASTPRT("For point Y( ), enter its coordinate value: y = ", 9, LCOL, 49,
RETURN.CODE%)
    CALL FASTPRT(" ", 10, LCOL, 49,
RETURN.CODE%)
    CALL FASTPRT("Remember to use consistent length units... ", 11, LCOL, 49,
RETURN.CODE%)

    CALL FASTPRT(STR$(IndexX) + SPACE$(1), 8, 24, 59, RETURN.CODE%)
    CALL FASTPRT(STR$(IndexY) + SPACE$(1), 9, 24, 59, RETURN.CODE%)

    CALL FASTPRT(LJustStr$(x(IndexX)) + SPACE$(6 - LEN(LJustStr$(x(IndexX)))), 8, 63,
10, RETURN.CODE%)
    CALL FASTPRT(LJustStr$(y(IndexY)) + SPACE$(6 - LEN(LJustStr$(y(IndexY)))), 9, 63,
10, RETURN.CODE%)

DO
    SELECT CASE ExitCode

        CASE 20

```



```

FOR I = IndexX TO nx
  CALL FASTPRT(STR$(I) + SPACE$(1), 8, 24, 59, RETURN.CODE%)
  NewGetIn LJustStr$(x(I)), 1, 6, 8, 63, 10, x$, ab$

  IF ab$ = CHR$(ESC) AND I = 1 THEN
    ExitCode = 666
    PCOPY 1, 0
    EXIT SUB
  ELSEIF ab$ = CHR$(ESC) THEN
    I = I - 2
  ELSEIF Filter(x$, "-1234567890.", "N") <> "" THEN
    ERROR 254
    x$ = ""
    I = I - 1
  ELSE
    x(I) = VAL(x$)
  END IF
NEXT I
ExitCode = 22

```

CASE 22

```

FOR J = IndexY TO ny
  CALL FASTPRT(STR$(J) + SPACE$(1), 9, 24, 59, RETURN.CODE%)
  NewGetIn LJustStr$(y(J)), 1, 6, 9, 63, 10, y$, ab$
  IF ab$ = CHR$(ESC) AND J = 1 THEN
    CALL FASTPRT(LJustStr$(y(J)), 9, 63, 10, RETURN.CODE%)
    ExitCode = 20
    IndexX = nx
    EXIT FOR
  ELSEIF ab$ = CHR$(ESC) THEN
    J = J - 2
  ELSEIF Filter(y$, "-1234567890.", "N") <> "" THEN
    ERROR 254
    y$ = ""
    J = J - 1
  ELSE
    y(J) = VAL(y$)
    'PRINT #3, y(J);
  END IF
NEXT J

IF J = 1 THEN
  ExitCode = 20
ELSE
  'PRINT #3,
  ExitCode = 900
END IF

```

END SELECT

```

LOOP UNTIL ExitCode = 900
PCOPY 1, 0
END SUB

```

```

DECLARE SUB PopDira (SEARCH$, SHOWITEMS%, FORE%, BACK%, HFORE%, HBACK%, QUADRANT$,
SHADOW%, NUMFILES%, SELECTFILE$, RETURN.CODE%)
'S$INCLUDE: 'CURRENTB.BI'
'Block INFILE works with POPUPS.LIB and GETIN.BAS

LocalHandler:
GlobalHandler
RESUME NEXT

REM $DYNAMIC
SUB InFile (ExitCode AS INTEGER)

    ON ERROR GOTO LocalHandler
    Footer
    CALL MAKEWIND(2, 3, 18, 78, 2, 11, 1, GROW, NOSHADOW, "", RETURN.CODE%)
    CALL MAKEWIND(4, LCOL - 2, 5, RCOL + 2, 2, 1, 11, GROW, SHADOW, "", RETURN.CODE%)

    IF Caller$ = "geometry" THEN
        CALL FASTPRT("Select the name of the file with the 3D geometry you want", 4,
        LCOL, 49, RETURN.CODE%)
        CALL FASTPRT("to read ...      Use the cursor keys and press <enter>      ", 5,
        LCOL, 49, RETURN.CODE%)
    ELSEIF Caller$ = "edit" THEN
        CALL FASTPRT(" Select the name of the data file you would like to edit ", 4,
        LCOL, 49, RETURN.CODE%)
        CALL FASTPRT("      Use the cursor keys and press <enter>      ", 5,
        LCOL, 49, RETURN.CODE%)
    END IF

    DO
        CALL PopDira("*.*", 6, 0, 11, 0, 11, "0", 1, numf%, Nam$, RETURN.CODE%)

        IF Nam$ = "" THEN
            ExitCode = 666
            EXIT SUB
        END IF

        OPEN Nam$ FOR INPUT AS #1

        '----- checking if file is legal -----
        LINE INPUT #1, Legal$

        IF Legal$ <> "PCSTBL3D Version 2.0" THEN
            ERROR 249
            ExitCode = 666
            CLOSE
        ELSE
            ExitCode = 999
        END IF

    LOOP UNTIL ExitCode <> 666

    COLOR 1, 7
    ClearScrn
    PCOPY 0, 3
    NotifyWindow "Please wait", "Reading data file " + Nam$

    '----- input nx, ny, ns -----
    INPUT #1, nx, ny
    INPUT #1, ns

    '----- enter X coordinates -----
    FOR I = 1 TO nx
        INPUT #1, x(I)
    NEXT I

    '----- enter Y coordinates -----
    FOR I = 1 TO ny
        INPUT #1, y(I)

```

```

NEXT I

'----- enter Z coordinates -----
FOR M = 1 TO ns
  FOR I = 1 TO nx
    FOR J = 1 TO ny
      INPUT #1, z(I, J, M)
    NEXT J
  NEXT I
NEXT M

'-----input surf generation data 1 -----
INPUT #1, xaprin
INPUT #1, yamin, dyamin, yamax, dyamax

'----- input surf generation data 2 -----
FOR I = 1 TO ny
  INPUT #1, xmin1(I), dxmin1(I), xmin2(I), dxmin2(I)
NEXT I

'----- input vertical constraints -----
FOR I = 1 TO nx
  FOR J = 1 TO ny
    INPUT #1, zmin(I, J), dzmin(I, J)
  NEXT J
NEXT I

'----- input soil properties -----
FOR I = 1 TO ns
  INPUT #1, Gama(I), C(I), Fi(I)
NEXT I

'----- pore pressure models -----
FOR K = 1 TO ns
  INPUT #1, utype(K)

  IF utype(K) = 1 THEN
    INPUT #1, ru
  ELSE
    FOR I = 1 TO nx
      FOR J = 1 TO ny
        INPUT #1, uz(I, J, K)
      NEXT J
    NEXT I
  END IF

NEXT K
CLOSE
PCOPY 3, 0

END SUB

REM $STATIC
'*****
'
'      SUB PROGRAM NAME          (PARAMETERS)          STATIC/RECURSIVE *
'-----
'
SUB PopDira (SEARCH$, SHOWITEMS%, FORE%, BACK%, HFORE%, HBACK%, QUADRANT$, SHADOW%,
NUMFILES%, SELECTFILE$, RETURN.CODE%) STATIC

      DEFINT A-Z                      'make all short interger by default

      RETURN.CODE% = 0
      DIR.RETURN.CODE = 0

```

```

'
' not sure that FILE$$ will be a FAR array, as it is a var. length string
'

REM      $DYNAMIC          'allocate array off far heap
      DIM FILE$(1)          'allocate an array of only 1 element
REM      $STATIC          'default arrays back to Basic data
segment

      FILES.MIN% = UBOUND(FILE$)          'this set LBOUND to OPTION BASE 1 also

      DIM SIZE.OF.FILE AS STRING * 10

POPDIR.START:
'      ATTR=&H00FF
      ATTR = NUMFILES%

      ATTR = ATTR AND &HFF

      COUNT = &HFF00 OR ATTR          'Get number of filenames that match
SEARCH$

      I = INT(VARPTR(FILE$(FILES.MIN%))) 'get pointer to start of array to hold
filenames found

      CALL DIR(SEARCH$, I, COUNT, DIR.RETURN.CODE) 'see how many files match our
filename spec.

      IF (COUNT = -1) OR (COUNT = 0) THEN          'were any matching filenames found?
          NUMFILES% = 0          'NO
          SELECTFILE$ = ""
          RETURN.CODE% = -2
          GOTO POPDIR.DONE
      END IF

      NUMFILES% = COUNT

      IF COUNT > 0 THEN          'allocate the filename array big
          REDIM FILE$(FILES.MIN% TO FILES.MIN% + COUNT) 'enough to hold all the
filenames
      END IF

      HEADERS$ = SEARCH$          'this was the filename spec used for
serach

      FOR I = FILES.MIN% TO FILES.MIN% + COUNT 'make each entry big enough to hold a
filename
          FILE$(I) = SPACE$(22)
      NEXT

      COUNT = ATTR

      I = INT(VARPTR(FILE$(FILES.MIN%)))

      CALL DIR(SEARCH$, I, COUNT, DIR.RETURN.CODE)

      FOR I = FILES.MIN% TO FILES.MIN% + COUNT
          SELECTFILE$ = FILE$(I)
          GOSUB EXPAND.FILES
          FILE$(I) = SELECTFILE$
      NEXT

      MAXITEMS = COUNT          'this many file names can be displayed
      SHOWFILES = SHOWITEMS%    'but only display this many at a time

      SELECT.% = FILES.MIN%

```

```

CALL POPLIST(HEADERS$, SHOWFILES, MAXITEMS, FILES$, FORE%, BACK%, HFORE%,
HBACK%, QUADRANT$, SHADOW$, SELECT.%, RETURN.CODE%)

IF SELECT.% = 0 THEN
    SELECTFILE$ = ""
ELSE
    SELECTFILE$ = LEFT$(FILES$(SELECT.%), 12)      'return the filename
selected
END IF

GOTO POPDIR.DONE

,
EXPAND.FILES:
    FILENAME$ = LEFT$(SELECTFILE$, 13)
    ASCIIIZ = INSTR(FILENAME$, CHR$(0))
    IF ASCIIIZ > 0 THEN
        FILENAME$ = LEFT$(FILENAME$, ASCIIIZ - 1)
    END IF

    FILENAME$ = FILENAME$ + SPACE$(80)

    FILE.ATTR$ = MID$(SELECTFILE$, 14, 1)
    FILE.ATTR = ASC(FILE.ATTR$)

    FILESIZE$ = ""
    FILEATTR$ = ""

    IF (FILE.ATTR AND &H10) <> 0 THEN
        FILESIZE$ = " <DIR> "
    END IF

    IF (FILE.ATTR AND &H8) <> 0 THEN
        FILEATTR$ = FILEATTR$ + "+ LABEL "
    END IF

    IF (FILE.ATTR AND &H2) <> 0 THEN
        FILEATTR$ = FILEATTR$ + "+ HIDDEN "
    END IF

    IF (FILE.ATTR AND &H4) <> 0 THEN
        FILEATTR$ = FILEATTR$ + "+ SYSTEM "
    END IF

    IF (FILE.ATTR AND &H1) <> 0 THEN
        FILEATTR$ = "+READONLY"
    END IF

    IF (FILE.ATTR AND &H20) <> 0 THEN
        FILEATTR$ = FILEATTR$ + "+ ARCHIVE"
    END IF

    MONTH$ = MID$(SELECTFILE$, 17, 1)
    MONTH = ASC(MONTH$) AND &HE0
    MONTH = MONTH / 32

    MONTH$ = MID$(SELECTFILE$, 18, 1)
    MONTH.TEMP = ASC(MONTH$) AND &H1
    MONTH.TEMP = MONTH.TEMP * 8

    MONTH = MONTH + MONTH.TEMP

    DAY$ = MID$(SELECTFILE$, 17, 1)
    DAY = ASC(DAY$) AND &H1F

    YEAR$ = MID$(SELECTFILE$, 18, 1)
    YEAR = ASC(YEAR$) AND &HFE
    YEAR = YEAR / 2
    YEAR = 1980 + YEAR

```

```

HOURS$ = MID$(SELECTFILE$, 16, 1)
HOURS = ASC(HOURS$) AND &HF8
HOURS = HOURS / 8

MINUTES$ = MID$(SELECTFILE$, 16, 1)
MINUTES = ASC(MINUTES$) AND &H3
MINUTES = MINUTES * 8

MINUTES$ = MID$(SELECTFILE$, 15, 1)
MINUTES.TEMP = ASC(MINUTES$) AND &HEO
MINUTES.TEMP = MINUTES.TEMP / 32
MINUTES = MINUTES + MINUTES.TEMP

SECONDS$ = MID$(SELECTFILE$, 15, 1)
SECONDS = ASC(SECONDS$) AND &H1F
SECONDS = SECONDS * 2

FILESIZE.ADDR = VARPTR(FILESIZE&)
POKE FILESIZE.ADDR, ASC(MID$(SELECTFILE$, 19, 1))

FILESIZE.ADDR = VARPTR(FILESIZE&)
POKE FILESIZE.ADDR + 1, ASC(MID$(SELECTFILE$, 20, 1))

FILESIZE.ADDR = VARPTR(FILESIZE&)
POKE FILESIZE.ADDR + 2, ASC(MID$(SELECTFILE$, 21, 1))

FILESIZE.ADDR = VARPTR(FILESIZE&)
POKE FILESIZE.ADDR + 3, ASC(MID$(SELECTFILE$, 22, 1))

IF FILESIZE$ = "" THEN
    FILESIZE$ = STR$(FILESIZE&)
    SIZE.OF.FILE = SPACE$(10)
    RSET SIZE.OF.FILE = FILESIZE$
    FILESIZE$ = SIZE.OF.FILE
END IF

FILEDATE$ = RIGHT$(STR$(MONTH), 2) + "-" + RIGHT$(STR$(DAY), 2) + "-" +
RIGHT$(STR$(YEAR), 2)

INSTR.POS = INSTR(2, FILEDATE$, " ")

IF INSTR.POS > 1 THEN
    MID$(FILEDATE$, INSTR.POS, 1) = "0"
END IF

FILETIME$ = RIGHT$(STR$(HOURS), 2) + ":" + RIGHT$(STR$(MINUTES), 2)

INSTR.POS = INSTR(2, FILETIME$, " ")

IF INSTR.POS > 1 THEN
    MID$(FILETIME$, INSTR.POS, 1) = "0"
END IF

MID$(FILENAME$, 13) = FILESIZE$
MID$(FILENAME$, 24) = FILEDATE$
MID$(FILENAME$, 34) = FILETIME$
MID$(FILENAME$, 40) = FILEATTR$

FILENAME$ = LEFT$(RTRIM$(FILENAME$), 12)

SELECTFILE$ = FILENAME$

RETURN

POPDIR.DONE:
ERASE FILE$
HEADERS$ = ""
'free memory allocated to array

```

```
FILENAME$ = ""  
FILESIZE$ = ""  
FILEATTR$ = ""  
FILETIME$ = ""  
FILEDATE$ = ""  
MONTH$ = ""  
DAY$ = ""  
YEAR$ = ""  
HOUR$ = ""  
MINUTE$ = ""  
SECOND$ = ""
```

```
END SUB
```


APPENDIX D   NEGATIVE NUMBERS FOR PRINTS

Figure	Film Number	Negative Number	Location of Negative
8	1	1	All negatives located at the following
9	1	3	address:
10	1	5	PURDUE UNIVERSITY
11	1	7	SCHOOL OF CIVIL ENGINEERING
12	1	9	West Lafayette, IN 47907
13	1	12	Room G245
14	1	17	(317) 494-5034
15	1	19	
16	1	24	
17	1	26	
18	1	30	
19	2	1	
20	2	4	
21	2	10	

COVER DESIGN BY ALDO GIORGINI