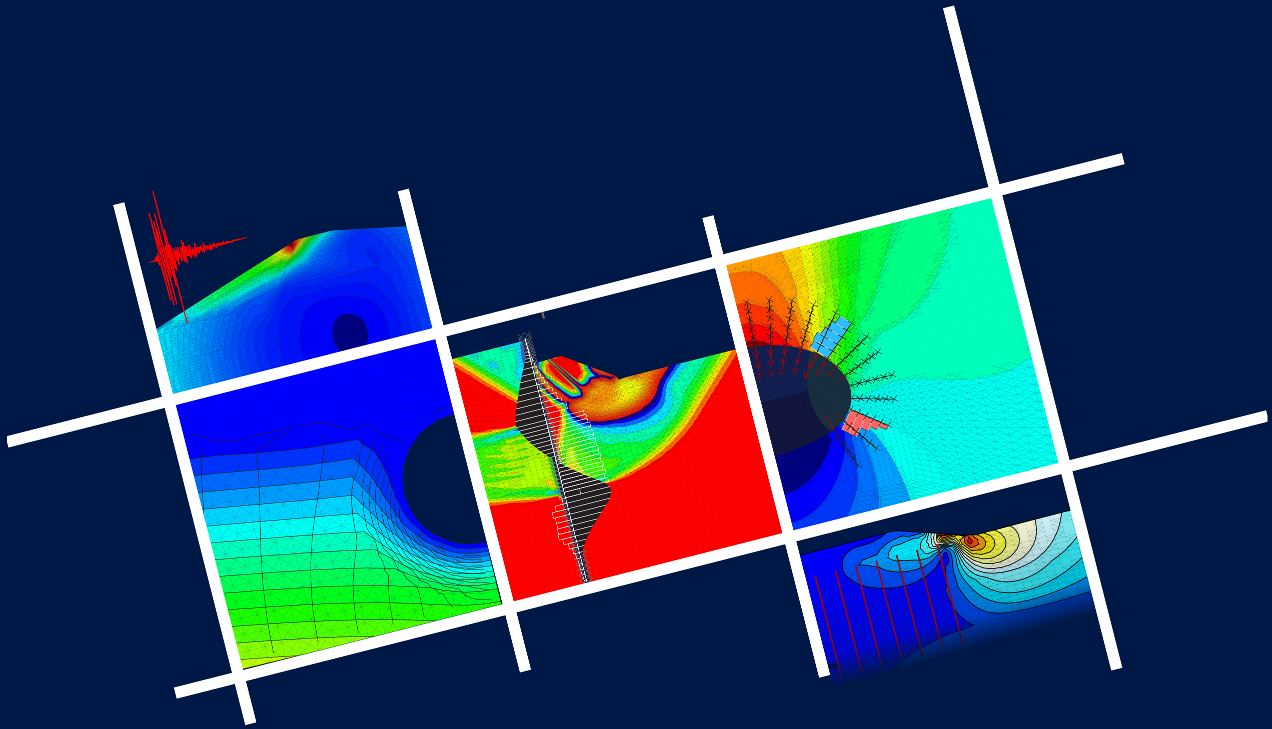
FLAC 8 Basics



An introduction to *FLAC* 8 and a guide to its practical application in geotechnical engineering

First Edition, Revision 01 (*FLAC*   Version 8.0) July 2015

Foreword

The inherent complexity associated with geotechnical problems and the lack of suitable analysis tools prompted the development of *FLAC* in 1986 by Itasca Consulting Group, Inc. Today, *FLAC* has over 4000 users and is applied in geo-engineering analyses by consultants, research organizations, universities and government agencies in more than 70 countries worldwide.

FLAC Basics, an introductory text on the application of *FLAC* in geotechnical engineering, was first written in 1993. This text was prepared as an introduction to the command-driven operation of *FLAC* to perform numerical analysis for geo-engineering problems. The last revision was prepared in 1998 and coincided with the release of *FLAC* Version 3.4. In 2000, a full graphical interface was added to *FLAC*, and *FLAC Basics* was discontinued.

This updated text, *FLAC 8 Basics,* is based on the original *FLAC Basics* and extends the documentation to demonstrate the power of the graphical interface and new features in version 8.0 to facilitate and enhance the operation of *FLAC* to solve complex geotechnical problems.

We have also included the foreword to the original 1993 edition of *FLAC Basics,* written by Charles Fairhurst, the founder of Itasca Consulting Group, Inc. We hope you appreciate this historical perspective.

Foreword to *FLAC Basics* (1993)

As the distinctive terms "rock mechanics," "soil mechanics," and "geomechanics" imply, the application of mechanics to engineering design in or on geological materials involves special considerations and a design philosophy different from that followed for design of fabricated materials. Designs for structures and excavations in or on rocks and soils must be achieved with relatively little site-specific data, and an awareness that deformability and strength properties may vary considerably, but to an unknown extent from place to place and with size of the structure. Joints, bedding planes, and other discontinuities and large-scale geological heterogeneities can have a major influence; the medium is "pre-loaded" by gravitational and tectonic forces, and may be subjected to transient forces imposed by rainfalls and earthquakes. Fluids under pressure carry part of the loads on the solid mass. Designs must therefore consider effects such as large and non-linear deformation behavior, including strain-softening. Controlled collapse of a structure, as in mining, may be a design objective. "Design as you go," the ability to modify a design as actual conditions are revealed during excavation, is often essential to successful, cost-effective completion of a project.

Given such considerations, it is not surprising that design procedures emphasizing linear, small strain, elasto-plastic behavior, and associated numerical approaches such as the implicit finite element method, used very successfully in other branches of engineering, have had less of an impact in geo-engineering.

In the late 1960s, Dr. Peter Cundall decided that there was merit in developing numerical modeling techniques better suited to the particular constraints of geotechnical design problems. His "distinct element method" based on an explicit finite difference numerical approach, allowed the discontinuous deformation of assemblages of blocky or particulate rock masses to be modeled, over unlimited deformations, to equilibrium or collapse. The two- and three-dimensional distinct element codes *UDEC* and *3DEC* are now widely used.

The same finite difference procedure has also been applied to develop the continuum code *FLAC*, which allows large and non-linear deformation, and other geotechnical features, to be easily modeled. *FLAC*, in its current version, is now used worldwide, and the three-dimensional version, *FLAC3D*, has recently been developed.

The codes are all designed for use on personal computers or workstations to allow the engineer to conduct numerical experiments, assessing the influence of possible variations from assumed conditions, as part of the design exercise.

Geoscientists and geotechnical engineers now have available codes well suited to their tasks, and there is no reason why advances, similar to those that have resulted from numerical modeling and the ongoing "computer revolution" in other fields of science and engineering, should not now occur for rock mechanics, soil mechanics, and geomechanics!

Charles Fairhurst
Minneapolis, November 1993

FLAC 8 Basics
Contents

Chapter 1 Welcome to *FLAC*

1.1	What is *FLAC*?	1
1.2	About this Guide	2
1.3	Range of Applications	2
1.4	Background and Validation	5
1.5	User Support	6

Chapter 2 Modeling and Methodology

2.1	Model Geo Engineering Processes	7
2.2	Explicit versus Implicit Solution	8

Chapter 3 Basic Operation Procedures

3.1	*FLAC* is Command driven	11
3.2	*FLAC* has a Graphical Interface	11
3.3	Start-Up	13
3.4	Before you Begin Modeling with *FLAC*	13
	3.4.1 *FLAC* Terminology	14
	3.4.2 Command Syntax	17
	3.4.3 Files	18
3.5	About The Finite Difference Grid	19
	3.5.1 Zones and Gridpoints	19
	3.5.2 (*i*,*j*) vs (*x*,*y*)	20
3.6	Some Notes on *FISH*	22
	3.6.1 Naming Conventions	22
	3.6.2 Scope of Variables	22
	3.6.3 Assignment of Data Types	23
	3.6.4 A Simple *FISH* Function	23

Chapter 4 A Simple Analysis: Failure of a Silty Clay Slope

4.1	Background	25
4.2	Solution Approach.	26
4.3	Model Description.	27
	4.3.1 Stage 1: Calculate FoS for a 45° Slope.	32
	4.3.2 Stage 2: Find Slope Angle for FoS = 1.3.	33
4.4	Results	38
4.5	On Your Own.	39

Chapter 5 Adding Complexity: A Shallow Tunnel in Soft Ground

5.1	Background.	40
5.2	Problem Description.	44
5.3	Modeling Procedure.	44
	5.3.1 Start-Up Conditions.	45
	5.3.2 Position of Model Boundaries.	45
	5.3.3 Designing the Model Grid.	46
	5.3.4 Boundary Conditions.	53
	5.3.5 Material Model and Properties.	54
	5.3.6 Initial Conditions.	56
	5.3.7 Running to Equilibrium.	58
	5.3.8 Modeling a Lined Tunnel.	59
	5.3.9 Calculation of a Ground Reaction Curve in *FLAC*	60
	5.3.10 Lining the Tunnel.	62
	5.3.11 Surface Settlement Profile	66
5.4	On Your Own.	67

Appendix A Conventions 68

Appendix B - Systems of Units 69

Figures

Figure 1.1 Stability analysis of contours in a benched slope. 3
Figure 1.2 Design calculation for a sheetpile supported excavation. 3
Figure 1.3 Simulation of a multi-stage tunnel excavation. 4
Figure 1.4 Evaluation of earthquake loading of a pile supported wharf. 4
Figure 2.1 Spectrum of modeling situation. 7
Figure 2.2 The basic explicit calculation cycle. 9
Figure 3.1 The graphical interface main window in *FLAC*. 12
Figure 3.2 Elements of a *FLAC* model 14
Figure 3.3 Identification of zone *i*,*j* indices. 19
Figure 3.4 Identification of gridpoint *i*,*j* indices. 20
Figure 3.5 (*i*,*j*) vs. (*x*,*y*) space. 21
Figure 4.1 Slope stability problem, solved analytically by Chen (2007), has a factor of safety of exactly 1.0 for a slope angle of $45^{o.}$ 25
Figure 4.2 Select "Include factor-of-safety calculations" to access the built-in factor ofsafety logic in *FLAC*. 26
Figure 4.3 Preference settings window. 27
Figure 4.4 Enter simple slope parameters using the [Build]/[Generate]/[Slope] tool. 27
Figure 4.5 Assign automatic boundary conditions. 28
Figure 4.6 Prescribe a "Medium" zoned mesh. 28
Figure 4.7 Define a Mohr-Coulomb material and properties. 29
Figure 4.8 Assign the material to all zones in the model by pressing [Set all]. 30
Figure 4.9 Apply gravitational acceleration 30
Figure 4.10 The model state is saved before performing the factor of safety calculation. 31
Figure 4.11 Perform a factor of safety calculation 32.
Figure 4.12 Factor of safety result for [Medium] zoning (FoS = 1.01). 32
Figure 4.13 Factor of safety result for [Extra fine] zoning (FoS = 1.00). 33
Figure 4.14 Create a 55 x 30 zone grid to be adjusted by the "slope_angle.fis" *FISH* function. 36
Figure 4.15 Create "Slope_angle.fis" in the *FISH* editor pane 36
Figure 4.16 Select [Parameters] to add *FISH* input parameters in the Fish editor. 37
Figure 4.17 Execute the "slope_angle.fis" *FISH* function and change input for LXtop RXtop, and fos_limit in the *FISH* input dialog. 37
Figure 4.18 The slope angle corresponding to FoS = 1.3 is printed in the Console output. 38
Figure 4.19 Factor of safety result (FoS = 1.3). 39
Figure 5.1 Geometry of tunnel and surface trough (Yeats, 1985). 41
Figure 5.2 Longitudinal displacement profile (Vlachopoulos and Diederichs, 2009). 42
Figure 5.3 Ground reaction curve shown with a support reaction curve for a liner installed at the tunnel face, and the relation to an LDP (Vlachopoulos and Diederichs, 2009). 43
Figure 5.4 Problem geometry and material properties. 44
Figure 5.5 Select "Include structureal elements" to access the structural element logic in *FLAC* 45
Figure 5.6 Influence of mesh size on geometric representation. 46
Figure 5.7 [BUILD] tool tabs. 47
Figure 5.8 [GENERATE] tool geometries menu. 47
Figure 5.9 Draw the outer model boundary with the [Box] edit mode. (Xmin = -60.0, Ymin = -45.0, Xmax = 60.0, Ymax = 0.0). 48
Figure 5.10 Draw the tunnel periphery with the [Circle] edit mode. (Xc = 0.0, Yc = -15.0, Radius = 3.0, Segments = 48). 49
Figure 5.11 Draw the gravel-clay boundary with the [Line] edit mode. (X1 = -60.0, Y1 = -11.0, X2 = 50.0, Y2 = 0.0). 49
Figure 5.12 Add construction lines to divide the model into quadrilateral blocks. 50

Figure 5.13 Select the [Boundary] edit stage of the [Edit] tool. 51
Figure 5.14 Reduce the number of attached lines in the model by setting attached lines around the tunnel region. 51
Figure 5.15 Create automatic zoning with the “Fine (100)” zoning option. 52
Figure 5.16 Adjust zoning manually with [Zone size (manual)] and [Adjust ratios] Edit modes. 53
Figure 5.17 Set boundary conditions in the [Boundary] edit stage of the [Edit] tool. 54
Figure 5.18 Create materials and select properties in the Define Material dialog. 55
Figure 5.19 Assign gravel and clay materials and properties in the [Materials] edit stage. 55
Figure 5.20 Intial geometry of *FLAC* model. 56
Figure 5.21 Stress distribution is initialized using the [In Situ]]/[Initial] tool. 57
Figure 5.22 Gravitational loading is assigned using the [Settings]/[Gravity] tool. 58
Figure 5.23 Solve for initial equilibrium using the [Run]/[Solve] tool and check “Solve initial equilibrium as elasic model”. 58
Figure 5.24 Plot maximum unbalanced force to check equilibrium using the [Plot]/[Quick] tool. 59
Figure 5.25 Plot gridpoint ID numbers in the [Plot]/[Model] tool and use the *FISH* editor to create *FISH* function vclos to create variables vlcos and hclos to monitor vertical and horizontal tunnel closure. 60
Figure 5.26 Apply reaction forces around the tunnel periphery in the [In Situ]/[Apply] tool. 61
Figure 5.27 Ground reaction curves showing normalized internal pressure in tunnel versus vertical and horizontal tunnel closure. 61
Figure 5.28 Ground reaction curves at 40% traction relaxation. ([Fix(Lock) View] in the View tool bar is pressed to lock the axes to match Figure 5.27.) 62
Figure 5.29 Connect liner elements to gridpoints along the tunnel periphery. 63
Figure 5.30 Continue relaxing forces around the tunnel periphery using the [In Situ]/[Apply] tool. 64
Figure 5.31 Vertical displacement of the soil and axial loads in the tunnel lining. 64
Figure 5.32 Vertical displacement of the soil and moments in the tunnel lining. 65
Figure 5.33 Moment-thrust diagram for shotcrete liner. 65
Figure 5.34 Shear-thrust diagram for shotcrete liner. 66
Figure 5.35 Settlement profile comparing *FLAC* results to empirical formulation. 67
Figure A.1 Sign convention for shear stress and shear strain. 68

Tables

Table 1.1 *FLAC* Development History 6
Table 2.1 Comparison of Explicit and Implicit Solution Methods 10
Table 3.1 Contents of “\\itasca*FLAC* 800\datafiles” Directories 13
Table 3.2 *FLAC* File Types 18
Table B.1 Systems of units - mechanical properties 69
Table B.2 Systems of units - groundwater flow parameters 69
Table B.3 Systems of units - structural elements 69

Chapter 1
Welcome to *FLAC*

1.1 What is *FLAC*?

F - **F**ast

L - **L**agrangian

A - **A**nalysis (of)

C - **C**ontinua

FLAC (Fast Lagrangian Analysis of Continua) is a two-dimensional, explicit finite difference numerical program for engineering mechanics computation. It was first developed in 1986 specifically to perform analyses on microcomputers operating on Microsoft Windows systems. Today, the software is designed to take advantage of multi-core processing for high-speed computation of model grids containing several thousand elements. Typical engineering problems were solved in several hours using the original *FLAC*. With the current *FLAC*, the solution time has reduced considerably. *FLAC* was originally developed for geotechnical and mining engineers, and since then, this versatile program has become an essential analysis and design tool in a variety of civil, mining, and mechanical engineering fields.

FLAC offers a wide range of capabilities to solve complex problems in mechanics. Materials are represented by elements within a grid that is adjusted by the user to fit the shape of the object to be modeled. Each element behaves according to a prescribed linear or non-linear stress/strain law in response to applied forces or boundary restraints. The material can yield and flow, and the grid can deform (in large strain mode) and move with the material that is represented. *FLAC* is based on a "Lagrangian" calculation scheme that is well suited for modeling large distortions and material collapse. Several built-in constitutive models are available to simulate highly non-linear, irreversible responses that are representative of geologic or similar materials.

In addition, *FLAC* contains many special features that extend usefulness including:

- interface elements to simulate distinct planes along which slip and/or separation can occur;
- groundwater and consolidation (fully coupled) models;
- plane strain, plane stress, and axisymmetric geometry modes;
- structural element models to simulate structural support (e.g., tunnel liners, rock bolts, geogrids, etc.);
- fully dynamic analysis capability;
- visco-elastic and visco-plastic (creep) models;
- thermal (and thermal-mechanical) modeling capability; and
- extensive facility for generating plots of virtually any problem variable in *FLAC*.

FLAC also contains a powerful built-in programming language, *FISH* (short for *FLAC*-ish), that enables the user to define new variables and functions. Users can write their own functions to extend *FLAC*'s usefulness and even implement their own constitutive models if so desired. *FISH* offers a unique capability to *FLAC* users who wish to tailor analyses to suit their specific needs.

1.2 About this Guide

This document presents an introductory guide to *FLAC*. It provides a brief overview of the range of *FLAC*'s applications and illustrates the power of this code as an analytical tool in geotechnical engineering. This guide also outlines the recommended procedure for applying *FLAC* to practical problems in geo-engineering. ***FLAC* 8 Basics** shows how to:

- create models (i.e., generate grids, specify boundary and initial conditions, define constitutive behavior and material properties);
- perform model alterations (e.g., excavate materials or change boundary conditions);
- reach a solution state (either a static equilibrium condition or a failure state);
- examine the model response (via plotted and printed output); and
- utilize the power of *FISH* to control and manipulate a *FLAC* model.

Two detailed example problems are provided to illustrate the procedure for *FLAC* modeling.

***FLAC* 8 Basics** is intended to assist engineers in making *FLAC* fit their own specific and complex problems, rather than making the problem fit the code. It complements the 14-volume *FLAC* 8 Manual and should be consulted as a guide to practical problem solving.

We recommend that new (and occasional) users first read Chapter 2 of the *FLAC* **User's Guide** for a beginner's guide and tutorial. Then, while working through the example problems in ***FLAC* 8 Basics** and becoming more experienced with running *FLAC*, look again to the ***User's Guide*** to learn how to use more of the features of this versatile computer program.

The remaining sections in Chapter 1 of ***FLAC* 8 Basics** provide an overview of some applications of *FLAC* and a history of the code's development. A recommended methodology for using *FLAC* to solve problems involving geo-engineering processes is detailed in Chapter 2. This chapter also explains the solution algorithm used in *FLAC*. Chapter 3 contains basic operating procedures for running *FLAC* and presents an overview of the terminology in the code. Two practical applications are described in the last two chapters, including a simple slope stability analysis and then a more complex problem involving a shallow tunnel in soft ground. These are typical problems in soil mechanics and serve to demonstrate the power of *FLAC* in geotechnical analysis.

1.3 Range of Application

Problems in geotechnical engineering encompass a wide range of physical processes. The power of *FLAC* is its ability to simulate these processes either individually or in combination. Mechanical, fluid flow, and thermal analyses can be performed as separate or coupled calculations. For example, saturated- and unsaturated-groundwater flow modeling can be performed independently or coupled to the mechanical stress calculation. Likewise, heat transfer calculations can be run alone or coupled to thermal stress calculations. Models can be run to a static equilibrium solution and then subjected to dynamic excitations. Structural element calculations can be run independently or coupled to the *FLAC* grid. The degree to which the analyses are integrated is the prerogative of the user, and the couplings can be turned on and off at the user's discretion.

In this way, *FLAC* can be readily applied to both simple and complex analyses. The user decides the level of complexity to be modeled. For example, *FLAC* models can be created for simple mechanical and effective stress calculations as well as for complex analyses involving coupled fluid/stress interaction and dynamic pore pressure change. Some example applications are illustrated in Figures 1.1, 1.2, 1.3, and 1.4.

Multiple local stability surfaces identified by factor of safety contours.

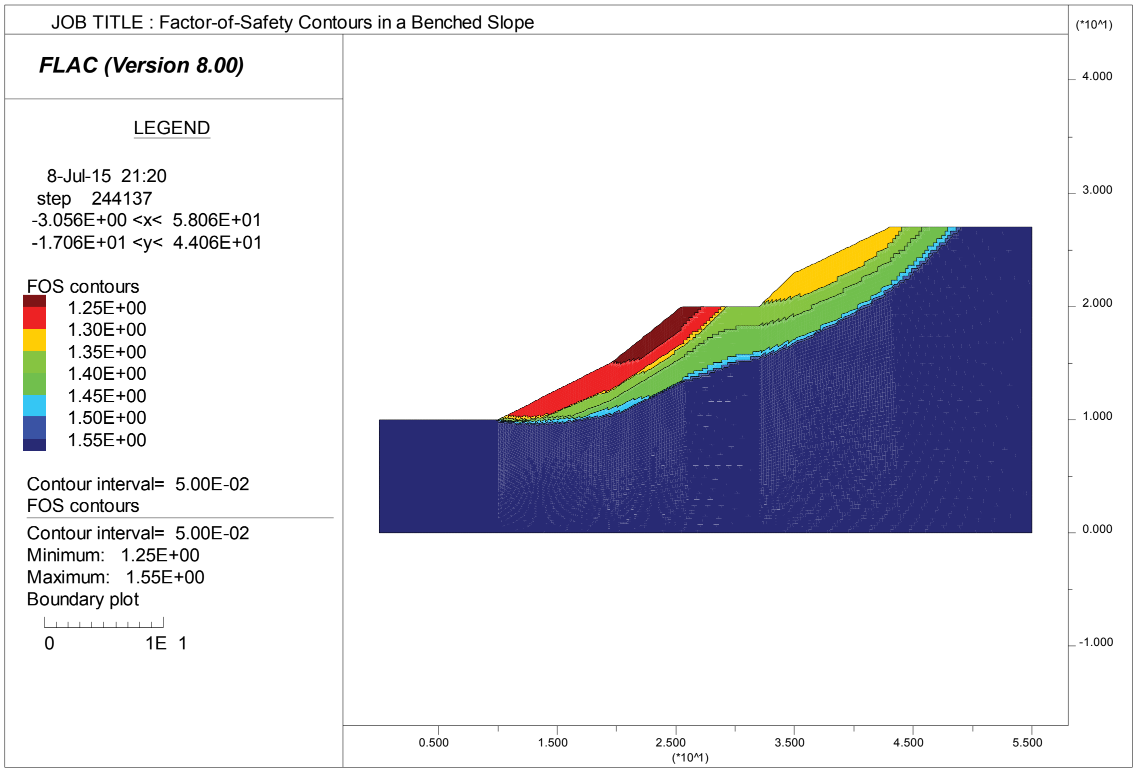


Figure 1.1 Stability analysis of a benched slope.

Calculation of sheetpile moments and forces and identification of multiple failure mechanisms from shear strain-rate contours.

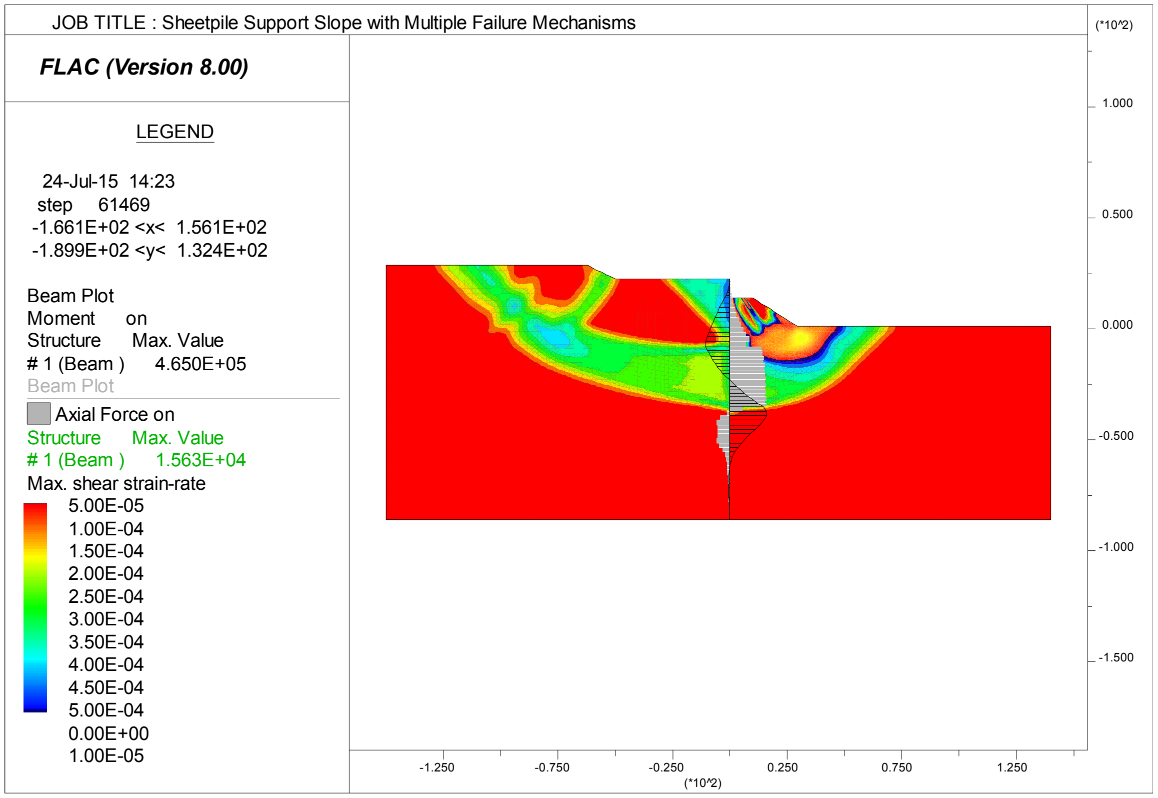


Figure 1.2 Design calculation for a sheetpile supported excavation.

Axial forces in rockbolts and displacements in the rock.

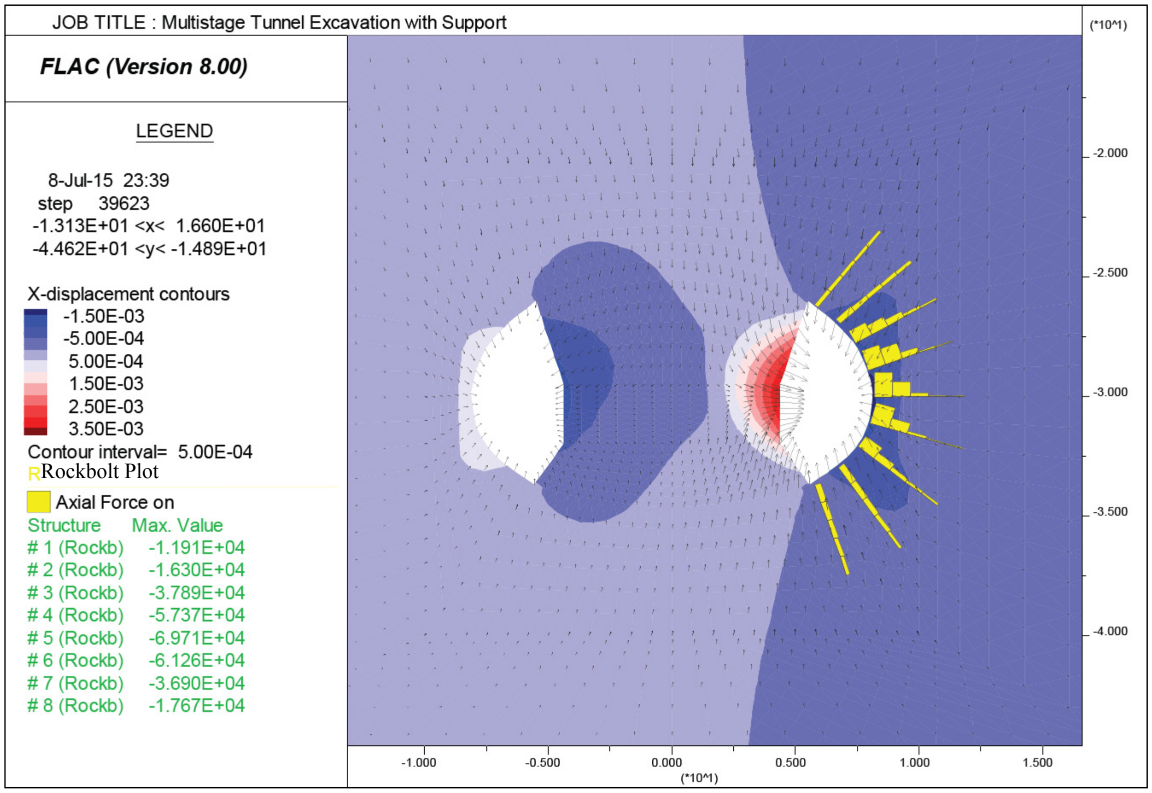


Figure 1.3 Simulation of a multi-stage tunnel excavation.

Deformation of an embankment and wharf structure resulting from earthquake excitation (large strain simulation).

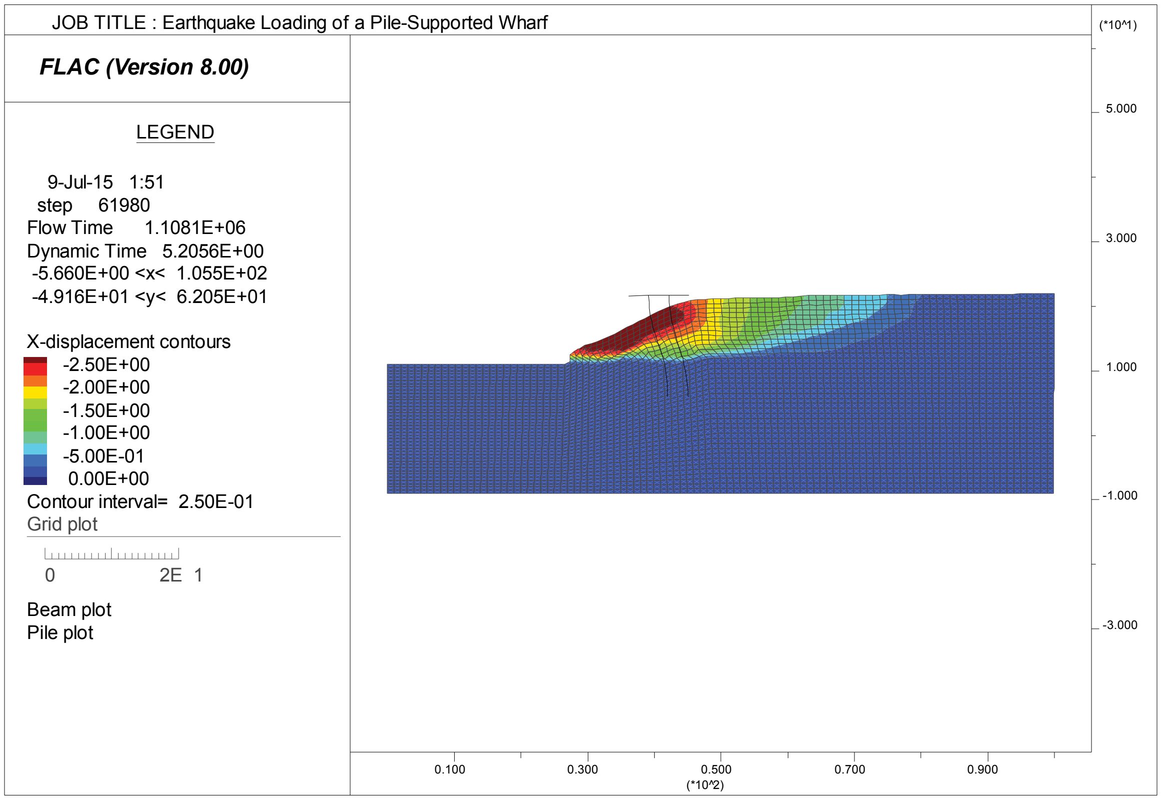


Figure 1.4 Evaluation of earthquake loading of a pile-supported wharf.

Types of processes that can be treated in *FLAC* and typical problem applications are:

- mechanical loading capacity and deformations (slope stability and foundation design);
- evolution of progressive failure and collapse (hard rock mine and tunnel design);
- time-dependent creep behavior of viscous materials (salt and potash mine design);
- restraint provided by structural support on geologic materials (tunnel lining, rock bolting, tiebacks, and soil nailing);
- saturated and unsaturated fluid flow and pore pressure buildup and dissipation for undrained and drained loading (reservoir engineering, groundwater flow, and consolidation studies of earth-retaining structures);
- coupled mechanical-fluid flow interaction (depletion of reservoirs);
- dynamic loading on slip-prone geologic features (earthquake engineering and mine rockburst studies);
- dynamic effects of explosive loading and vibrations (tunnel driving or mining operations);
- seismic excitation of structures such as dams and the coupled effect of fluid on time-dependent pore pressure change (liquefaction phenomena in foundations and dams); and
- deformation and mechanical instability resulting from thermal induced loads (performance assessment of underground repositories of high-level waste).

1.4 Background and Validation

Historically, Itasca only used *FLAC* in our mining consulting activities, but clients began to want access to our analysis tools. This spurred us to make *FLAC* available as a product outside our consulting, and it is now used around the world by civil and mining engineers, as well as in many educational and research institutions.

See the **Verification Problems** and **Example Applications** volumes for test examples.

The wide usage has led the demand for more features, and *FLAC* has grown apace. Since its first commercial release in February 1986, *FLAC* has undergone two major upgrades and several minor options have been added. Table 1.1 traces the development path.

FLAC has been tested and verified in a variety of problem settings. *FLAC* is actively used in our consultancy work, and as new problems arise, our own engineers work to develop *FLAC* to meet these modeling requirements. In this way, Itasca engineers and many of the *FLAC* users continue to add to the collection of verification problems and validation examples. The accuracy of the numerical formulation and solution scheme embodied in *FLAC* is comparable to that of other commercially available programs.

Table 1.1 *FLAC* **Development History**

Date	Version	Major Additions
February 1986	1.0	First commerical release of the PC version.
March 1987	2.0	Interface logic added.
November 1988	2.1	Thermal and creep options made available.
June 1989	2.2	Groundwater flow added.
September 1991	3.0	*FISH* added. Dynamic option made available.
November 1992	3.2	*FISH* consitutive modeling capability added.
April 1995	3.3	Pile elements and Cam Clay model added. New manual prepared.
September 1998	3.4	Windows-console version. Manual provided on CD-ROM
September 2000	4.0	Full graphical interface added.
May 2005	5.0	Speed up to groundwater flow calculation. New structural element types added (rockbolts, liners, strip reinforcement).
August 2008	6.0	Automatic rezoning logic made available for large strain calculations.
October 2011	7.0	Multithread mechanical calculations for faster runs on multi-core computers.
December 2015	8.0	64-bit version made available. Multithread fluid flow logic for faster flow calculations. Five new soil & rock constitutive models.

1.5 User Support

We believe that the support that Itasca provides to code users is a major factor in the popularity of our software. Itasca is more than just a developer and distributor of engineering software; our engineers also specialize in mining and civil engineering consulting and research. We have the background and expertise to answer specific questions related to geo-engineering analysis and design. Our support goes beyond answering basic hardware/software questions; we often assist users with engineering modeling practical applications.

We encourage you to contact us when you have a modeling question. We provide timely responses via telephone or email. General assistance in the installation of *FLAC* and answers to questions concerning capabilities of various features are provided free of charge. Technical assistance for specific user-defined problems can be purchased on an as-needed basis.

Chapter 2
Modeling Methodology

2.1 Modeling Geo-Engineering Processes

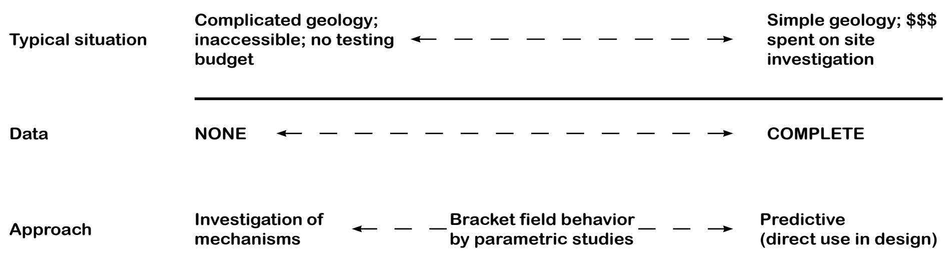
Can *FLAC* be used in design?

Starfield and Cundall (1988)[1] propose that the methodology of numerical modeling in geomechanics should be very different from that in other engineering fields, such as structural engineering. They point out that it is impossible, even in principle, to obtain complete field data at a rock or soil site. Information on stresses, properties, and discontinuities can only be partially known, at best. This situation is incompatible with the popular conception of the way in which computer programs are used in design (i.e., as a black box that accepts data input at one end and produces a prediction of behavior at the other). In contrast with this view, geomechanics programs should be used to discover mechanisms, especially when the input data necessary for prediction are absent. Once the behavior of the system is understood, it is then appropriate to use simple calculations in the design process. In other words, geomechanics programs may not be appropriate for use directly in design, but, more often, as experimental tools to help give the designer insight into mechanisms.

Unfortunately, the prescription summarized above is subject to misinterpretations. Some people use the following erroneous chain of "reasoning" when thinking about geomechanics modeling.

- Program X models geomechanical processes.
- Input data are always lacking in geomechanical processes, so accurate predictions cannot be made.
- Hence, Program X does not model accurately and can only be used qualitatively.

The fallacy here is that non-prediction should be associated with the field of application rather than with a particular computer program. The results of a computer program may be perfectly accurate when the program is supplied with appropriate data. Modelers should recognize that there is a continuous spectrum of situations, as illustrated in Figure 2.1 below.



Typical situation
Complicated geology; inaccessible; no testing budget
Simple geology; $$$ spent on site investigation
Data
NONE
COMPLETE
Approach
Investigation of mechanisms
Bracket field behavior by parametric studies
Predictive (direct use in design)

Figure 2.1 Spectrum of modeling stituation.

1. Starfield, A.M., and P.A. Cundall. "Towards a Methodolgy for Rock Mechanics Modelling," *Int. J. Rock Mech. Min. Sinc. & Geomech. Abstr.,* ***25***(3), 99– 106. (1998).

A program such as *FLAC* may either be used in a fully predictive mode (right-hand side of diagram) or as a numerical laboratory to test ideas (left-hand side). It is the field situation (and budget), rather than the program, that determines the types of use. But since *FLAC* is often applied to very complicated, non-linear systems, people tend to believe that the program can only be applied at the left-hand end of the spectrum. This is not true! If enough data are available, programs like *FLAC* can give good, accurate, and reliable predictions.

2.2 Explicit vs. Implicit Solution

How does *FLAC* differ from a finite element code?

We are often asked what the difference is between *FLAC* and a finite element code. *FLAC* is an explicit finite difference program. In the finite difference method, every derivative in the set of governing equations is replaced directly by an algebraic expression written in terms of the field variables (e.g., stress or displacement) at discrete points in space; these variables are undefined anywhere else.

In contrast, the finite element method has a central requirement that the field quantities (stress, displacement) vary throughout each element in a prescribed fashion using specific functions controlled by parameters. The formulation consists of adjusting these parameters to minimize error terms on local or global energy.

Both methods produce a set of algebraic equations to solve. Even though these equations are derived in quite different ways, it is easy to show (in specific cases) that the resulting equations are *identical* for the two methods. It is pointless to argue about the relative merits of finite differences or finite elements; the resulting equations are the same.

However, over the years, certain traditional ways of doing things have taken root. For example, finite element programs often combine the element matrices into a large global stiffness matrix. In contrast, this is not normally done with finite differences, because it is relatively efficient to regenerate the finite difference equations at each step. *FLAC* uses an explicit, time-marching method to solve the algebraic equations, while implicit, matrix-oriented solution schemes are more common in finite elements. Other differences are also common, but it should be stressed that features may be associated with one method rather than another because of habit more than anything else.

The numerical formulation and solution process are explained in detail in Section 1 of **Theory and Background**.

Even though we want *FLAC* to find a static solution to a problem, the dynamic equations of motion are included in the formulation. One reason for doing this is to ensure that the numerical scheme is stable when the physical system being modeled is unstable. With non-linear materials, there is always the possibility of physical instability; for example, the sudden collapse of a pillar. In real life, some of the strain energy in the system is converted into kinetic energy, which then radiates away from the source and dissipates. *FLAC* models this process directly because inertial terms are included—kinetic energy is generated and dissipated. In contrast, schemes that do not include inertial terms must use some numerical procedure to treat physical instability; the path taken may not be a realistic one. The penalty for including the full laws of motion is that the user must have some physical feel for what is going on. *FLAC* is not a black box that will give the solution. The behavior of the numerical system must be interpreted.

The general calculation sequence embodied in *FLAC* is illustrated in Figure 2.2. The equations of motion are first invoked to derive velocities and displacements from stresses and forces. Then, strain rates are derived from the velocities and new stresses from the strain rates. Take one timestep for every cycle around the loop. The important thing to realize is that each box in Figure 2.2 updates all of its grid variables from known values that remain fixed while control is within the box. For example, the lower box takes the set of velocities already calculated and computes new stresses for each element. The velocities are assumed to be frozen for the operation of the box (i.e., the newly calculated stresses do not affect the velocities). This may seem unreasonable, because we know that if a stress changes somewhere, it will influence its neighbors and change their velocities. However, a timestep is chosen that is so small that information cannot physically pass from one element to another in that interval. All materials have some finite speed at which information can propagate. Since one loop of the cycle occupies one timestep, the assumption of frozen velocities is justified. Frozen velocity is the concept that neighboring elements cannot affect one another during the period of calculation. Of course, after several cycles of the loop, disturbances can propagate across several elements, just as they would propagate physically.

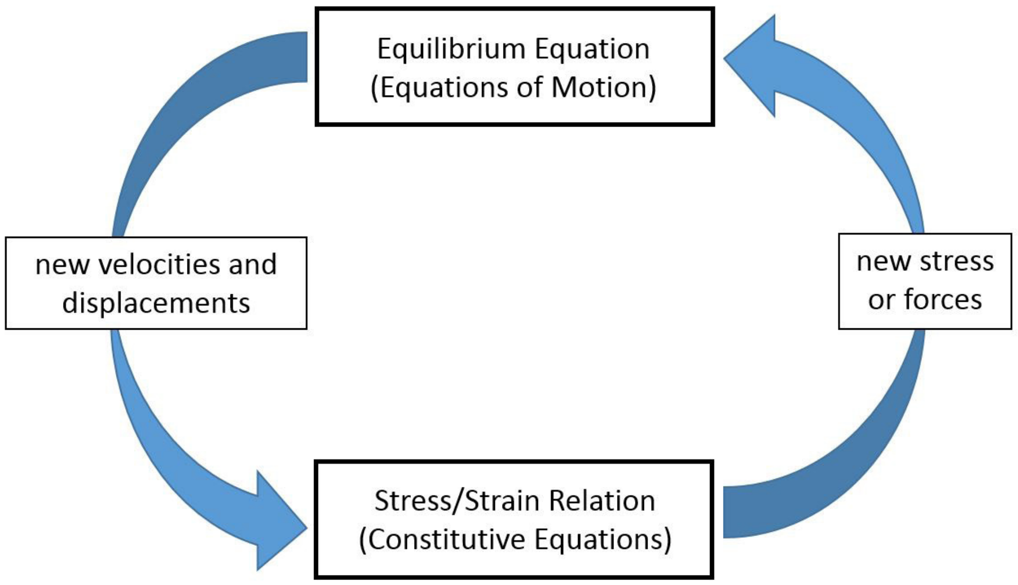


Figure 2.2 The basic explicit calculation cycle.

The central concept of the explicit method is that the calculation wave speed always keeps ahead of the physical wave speed; thus, the equations always keep ahead of the physical wave speed and operate on known values that are fixed for the duration of the calculation. There are several distinct advantages to this (and one big disadvantage) as listed in Table 2.1.

Table 2.1 Comparison of Explicit and Implicit Solution Methods

Explicit	Implicit
Timestep must be smaller than a critical value for stability.	Timestep can be arbitrarily large, with unconditionally stable schemes.
Small amount of computational effort per timestep.	Large amount of computational effort per timestep.
No significant numerical damping introduced for dynamic solution.	Numerical damping dependent on timestep present with unconditionally stable schemes.
No iterations necessary to follow nonlinear constitutive law.	Iterative procedure necessary to follow nonlinear constitutive law.
Provided that the timestep criterion is always satisfied, nonlinear laws are always followed in a valid physical way.	Always necessary to demonstrate that the above mentioned procedure is (a) stable, and (b) follows the physically correct path (for path-sensitive problem).
Matrices are never formed. Memory requirements are always at a minimum. No bandwidth limitations.	Stiffness matrices must be stored. Must find ways to overcome associated problems such as bandwidth.
Since matrices are never formed, large displacements and strains are accomodated without additional computing effort.	Additional computing effort needed to follow large displacements and strains.

Most importantly, no iteration process is necessary when computing stress from strains in an element, even if the constitutive law is wildly non-linear. In an implicit method (which is commonly used in finite element programs), every element communicates with every other element during one solution step, and several cycles of iteration are necessary before compatibility and equilibrium are obtained. The disadvantage of the explicit method is seen to be the small timestep, which means that large numbers of steps must be taken. Overall, explicit methods are best for ill-behaved systems (e.g., non-linear, large strain, physical instability); they are not as efficient for modeling linear, small-strain problems.

Chapter 3
Basic Operation Procedures

3.1 *FLAC* is Command Driven

FLAC is a command-driven computer program. Word commands control the operation of the program. This is an important distinction, especially if you are familiar with menu-driven software. We believe that the command-driven structure is better suited for conducting an engineering analysis for the following reasons.

- Engineering simulations usually consist of a lengthy sequence of operations, e.g., establish in-situ stress, apply loads, excavate tunnel, install support, and so on. A series of input commands (from a file or from the keyboard) corresponds closely with the physical sequence that they represent.
- A *FLAC* data file can be easily modified with a text editor. Several data files can be linked together to run a number of *FLAC* analyses in sequence. This is ideal for performing parameter studies.

The **CALL** command is used to call a data file into *FLAC*.

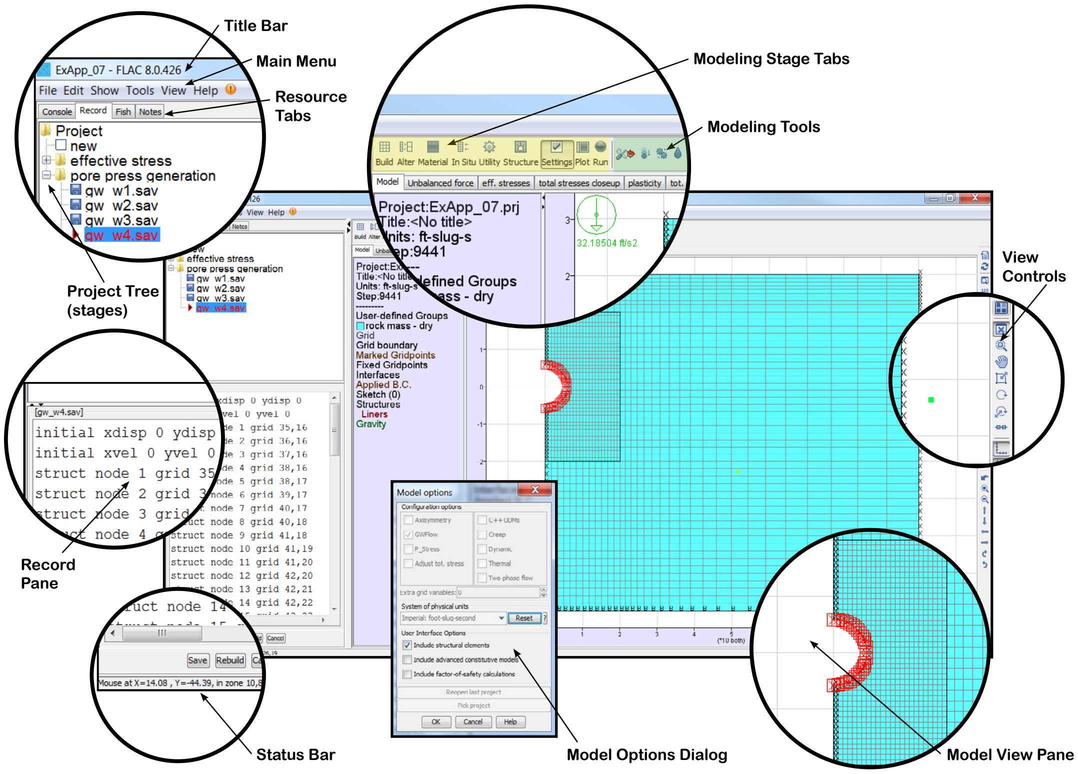
- The word-oriented input files provide an excellent means for keeping a documented record of the analyses performed for an engineering study.
- The command-driven structure allows you to develop pre- and post-processing programs to manipulate *FLAC* input/output as desired. For example, you may wish to write a mesh generation function to create a special grid shape for a series of *FLAC* simulations. This can readily be accomplished with the *FISH* programming language and incorporated directly in the input data file.

The input “language” is based on recognizable word commands that allow you to identify the application of each command easily and logically. The commands and data input are free format and can be entered through an “interactive” mode (i.e., via the keyboard) or “batch” mode (i.e., stored in a data file and read into *FLAC*).

The obvious drawback of the command-driven structure of *FLAC* is that you must learn the *FLAC* “language.” This can be especially frustrating for new or occasional users of *FLAC*. There are over 40 main commands and more than 400 command modifiers, called keywords, that are recognized by *FLAC*. To the new user, wading through all the commands to select those necessary for a desired analysis may seem an insurmountable task.

3.2 *FLAC* has a Graphical Interface

A menu-driven graphical interface is connected to the *FLAC* executable program to help new and occasional users of *FLAC* learn the *FLAC* language and recommended operation procedure. The graphical interface consists of three components: *Model-tool* panes, *Resource* panes, and *Model-view/plots* panes. The *Model-tool* panes are accessed from *Modeling-stage* tabs. The main window is shown in Figure 3.1. On startup, the main window displays two windows: one contains the *Resource* panes and the other contains a *Model-view* pane.



```
[gw_w4.sav]
initial xdisp 0 ydisp 0
initial xvel 0 yvel 0
struct node 1 grid 35,16
struct node 2 grid 36,16
struct node 3 grid 37,16
struct node 4 grid 38,16
struct node 5 grid 38,17
struct node 6 grid 39,17
struct node 7 grid 40,17
struct node 8 grid 40,18
struct node 9 grid 41,18
struct node 10 grid 41,19
struct node 11 grid 41,20
struct node 12 grid 42,20
struct node 13 grid 42,21
struct node 14 grid 42,22
```

Project:ExApp_07.prj
Title:<No title>
Units: ft-slug-s
Step:9441

Figure 3.1 The graphical interface main window in *FLAC*.

There are four tabbed *Resource* panes. The *Record* pane, shown in Figure 3.1, shows a record of *FLAC* commands associated with the current model project state. This pane can be edited and exported as a data file, and thus provides a list of all *FLAC* commands that represent the problem being analyzed. The *Record* pane also shows a “project tree” that displays a tree list of the saved states created for a model. The *Console* pane shows the text output and allows command line input at the bottom of the pane. The third *Resource* pane is the *Fish editor*. *FISH* functions can either be created or called into the *FLAC* model using the *Fish editor* pane. The final *Resource* pane is the *Notes* pane, which provides a means to create a text record describing the model.

The *Model view* pane shows a graphical view of the model. Additional plot views can be added as tabbed panes in this window; these views display user-created plots.

There is a tab set above the model-view pane that contains toolbars for each of the modeling stages: [BUILD], [ALTER], [MATERIAL], [IN SITU], [UTILITY], [SETTINGS], [PLOT], and [RUN].

The [STRUCTURE] tab also appears in the tab set when this option is checked in the *Model options* dialog box. By clicking on one of the tabs, a pane is activated containing tools that provide the necessary controls to create and run your model.

The *Model Options* dialog, shown in Figure 3.1, appears every time you start *FLAC* or begin a new model project. The dialog identifies which modes of analysis are available to you in your version of *FLAC*.

3.3 Start-Up

The default installation procedure creates an [Itasca] group under [Programs] on the user's [Start] menu in Windows. The [Itasca] group contains the [*FLAC*]—>[*FLAC* 8.00] shortcut. After the installation is complete, the *FLAC* hardware key should be attached to the USB port on your computer.

To load *FLAC*, simply click on the [*FLAC* 8.00] icon. When loaded, the *FLAC* window appears, as shown in Figure 3.1. When *FLAC* is loaded for the first time, a dialog that requests permission to copy data files and other user resources to your documents folder will appear. This is done to avoid permission conflicts in the operating system when attempting to open files in the "C:\Program Files" folder. It is recommended that you allow *FLAC* to copy the files to your "My Documents\itasca\FLAC 800" folder.

Please note the files contained in the "My Documents\itasca\FLAC 800\datafiles" directory. This directory has several sub-directories that contain files that can be executed in *FLAC* to demonstrate various capabilities and features. Table 3.1 shows a list of these sub-directories and a description of their contents.

Table 3.1 Contents of "\\itasca*FLAC* 800\datafiles" Directories

Directory	Description
ConstitutiveModels	Mechanical constitutive model exercises
Creep	Creep material model exercises
Dynamic	Dynamic analysis exercises
ExampleApplications	Practical example application problems
Fish	*FISH* in *FLAC* exercises
FLAC_Basics	***FLAC* 8 Basics** examples
FLAC_Slope	*FLAC* Slope FOS examples
Fluid	Fluid-mechanical interaction exercises
FOS	Factor of safety exercises
Structures	Structural element exercises
Theory	*FLAC* theory and background exercises
Thermal	Thermal and coupled thermal exercises
UsersGuide	*FLAC* introductory exercises
VerificationProblems	Verification problems for *FLAC*

3.4 Before You Begin Modeling with *FLAC*

There are a few things to know before creating and running a *FLAC* model.

- You need to understand the *FLAC* terminology. This includes nomenclature and the numbering system for defining the elements of a *FLAC* model.
- You need to know the syntax for the *FLAC* input language.
- You need to be aware of the different types of files that are used and created by *FLAC*.

The following sections explain each of these aspects in detail.

3.4.1 *FLAC* Terminology

The nomenclature used in *FLAC* is generally consistent with that in conventional finite difference or finite element programs that perform stress analysis. Figure 3.2 illustrates the terms described here.

Each zone is further subdivided into two overlayed sets of constant strain triangular elements. This is done to provide accurate solution for problems in plasticity. For more details, see Section 1 of **Theory and Background**.

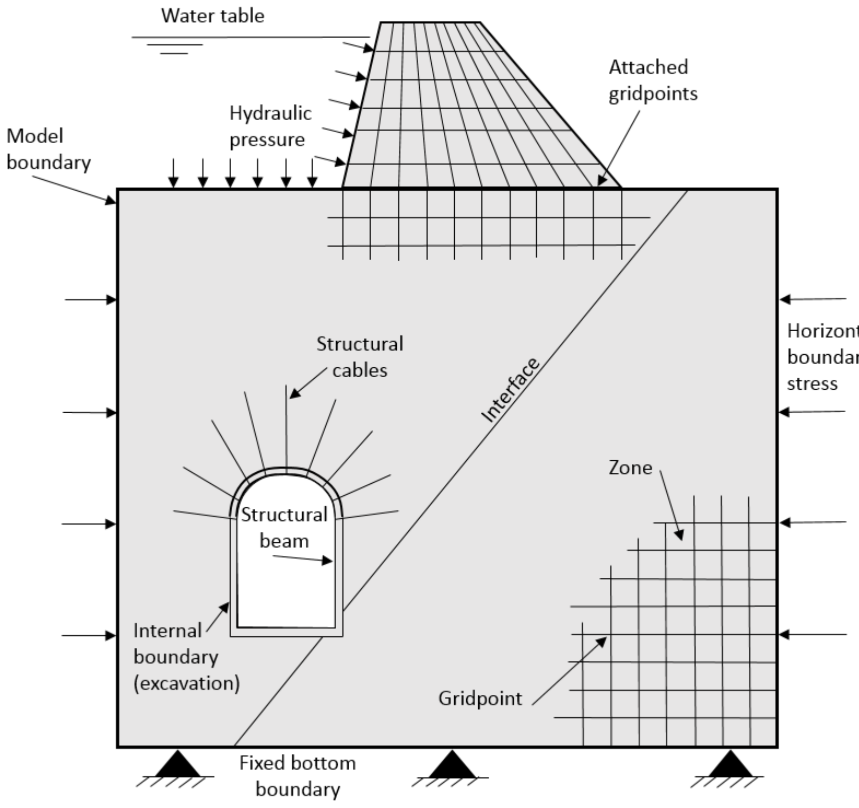


Figure 3.2 Elements of a *FLAC* model.

***FLAC* MODEL**	The *FLAC* model is created by the user to simulate a physical problem. When referring to a *FLAC* model, you are implying a sequence of *FLAC* commands that describe the problem conditions for numerical solution.
ZONE	The finite difference zone is the smallest geometric domain within which the change in a phenomenon (e.g., stress/strain, fluid flow, or heat transfer) is calculated. In *FLAC*, the model domain is divided into quadrilateral zones. Another term for “zone” is “element.”
GRIDPOINT	Gridpoints are associated with the corners of the finite difference zones. There are always four gridpoints associated with each zone. A pair of *x* and *y* coordinates are defined for each gridpoint, thus specifying the exact location of the finite difference zone. Another term for “gridpoint” is “nodal point,” or “node.”

GROUP	A group is a collection of zones identified by a unique name. Groups are used to limit the range of certain *FLAC* commands, such as the **MODEL** command that assigns material models to designated groups of zones.
FINITE DIFFERENCE GRID See Section 3.5 for more details.	The finite difference grid is an assemblage of one or more finite difference zones across the physical region being analyzed. Another term for "grid" is "mesh." The finite difference grid also identifies the storage locations of all state variables in the model. The procedure followed in *FLAC* is that all vector quantities (e.g., forces, velocities, displacements) are stored at gridpoint locations, while all scalar and tensor quantities (e.g., stresses, pressure, material properties) are stored at zone centroid locations. There are four exceptions: saturation and temperature are considered gridpoint variables; pore pressure is stored at both gridpoint and zone centroid locations; and fluid flow rate is considered a zone variable.
MODEL BOUNDARY	The model boundary is the periphery of the finite difference grid. Internal boundaries (i.e., holes within the grid) are also model boundaries.
BOUNDARY CONDITION	A boundary condition is the prescription of a constraint or controlled condition along a model boundary (e.g., a fixed displacement or force for mechanical problems, an impermeable boundary for groundwater flow problems, or adiabatic boundary for heat transfer problems).
INITIAL CONDITIONS	This is the state of all variables in the model (e.g., stresses and pore pressures) prior to any loading change or disturbance (e.g., excavation).
SUB-GRID	The finite difference grid can be divided into sub-grids. Sub-grids can be used to create regions of different shapes in the model (e.g., the dam sub-grid or the foundation sub-grid in Figure 3.2).
NULL ZONES	Null zones are zones that represent voids (i.e., no material present) within the finite difference grid. All newly created zones are null by default.

CONSTITUTIVE MODEL The *FLAC* constitutive models, along with examples of representative materials and applications, are summarized in Table 3.2 in Section 3 of the **User's Guide**.	The constitutive (or material) model represents the deformation and strength behavior prescribed to the zones in a *FLAC* model. Several constitutive models are available in *FLAC*, simulating different types of behavior commonly associated with geologic materials. Constitutive models and material properties can be assigned individually to every zone in a *FLAC* model.
ATTACHED GRIDPOINTS	Attached gridpoints are pairs of gridpoints that belong to separate, but joined, sub-grids. In Figure 3.2, the dam is joined to the foundation along attached gridpoints. Attached gridpoints cannot separate from one another once attached.
INTERFACE	An interface is a connection between sub-grids that can separate (i.e., slide or open). An interface can represent a physical discontinuity such as a fault or contact plane. It can also be used to join sub-grids.
MARKED GRIDPOINTS	Marked gridpoints are specially designated gridpoints that delimit a region for the purpose of applying an initial condition, assigning a material model and properties, or printing selected variables. The marking of gridpoints has no effect on the solution process.
REGION	A region in a *FLAC* model refers to all zones enclosed within a contiguous string of "marked" gridpoints. Regions are used to limit the range of certain *FLAC* commands, such as the **MODEL** command, which assigns material models to designated regions.
STRUCTURAL ELEMENT	Structural elements are generally linear elements used to represent the interaction of structures (such as tunnel liners, rock bolts, cable bolts, or support props) with a soil or rock mass. Non-linear effects are possible with cable elements, rock bolts, or support elements.
STEP When using the dynamic analysis option in *FLAC*, **STEP** refers to the actual timestep for the dynamic problem.	Since *FLAC* is an explicit code, the solution to a problem requires a number of computational steps. During computational stepping, the information associated with the phenomenon under investigation is propagated across the zones in the finite difference grid. A certain number of steps is required to arrive at an equilibrium (or steady flow) state for a static solution. Typical problems are solved within 2000 to 4000 steps, although large complex problems can require tens of thousands of steps to reach a steady state. Other terms for "step" are "timestep" and "cycle."

STATIC SOLUTION

A static or quasi-static solution is reached in *FLAC* when the rate of change of kinetic energy in a model approaches a negligible value. This is accomplished by damping the equations of motion. At the static solution stage, the model will be either at a state of force equilibrium or at a state of steady flow of material if a portion (or all) of the model is unstable (i.e., fails) under the applied loading conditions. This is the default calculation mode in *FLAC*. Static mechanical solutions can be coupled to transient groundwater flow or heat transfer solutions. (As an option, fully dynamic analysis can also be performed by inhibiting the static solution damping.)

UNBALANCED FORCE

The maximum nodal force vector is also called the "unbalanced" or "out-of-balance" force**.**

The unbalanced force indicates when a mechanical equilibrium state (or the onset of plastic flow) is reached for a static analysis. A model is in exact equilibrium if the net nodal force vector at each gridpoint is zero. The maximum nodal force vector is monitored in *FLAC* and printed to the screen when the **STEP** or **SOLVE** command is invoked. The maximum unbalanced force will never exactly reach zero for a numerical analysis. The model is considered to be in equilibrium when the maximum unbalanced force is small compared to the total applied forces in the problem. Failure and plastic flow occurring within the model is indicated when the unbalanced force approaches a constant non-zero value.

3.4.2 Command Syntax

All input commands are word oriented and consist of a primary command word followed by one or more keywords and values, as required. Some commands accept switches, that is, keywords, that modify the action of the command. Each command has the following format:

COMMAND **key**word value ...< **key**word value ...>

Optional parameters are denoted by < >, while the ellipsis (...) indicates that an arbitrary number of such parameters may be given. The commands are typed literally on the command line. You will note that only the first few letters are in bold type. The program requires only these letters to be typed for the command to be recognized, and is not case sensitive. The entire word for commands and keywords may be entered if the user so desires.

Many of the keywords are followed by a series of values that provide the numeric input required by the keyword. Values identified with *i*, *j*, *m*, or *n* indicate that an integer value is expected; otherwise, a real (or decimal) value is required. The decimal point may be omitted from a real value but may not appear in an integer value.

Commands, keywords, and numeric values may be separated by any number of spaces or by any of the following characters:

( ) , =

A semicolon may be used to precede comments; anything in the input line after a semicolon is ignored. An input line, including comments, may contain up to 200 characters.

Operations using the graphical interface will result in the creation of *FLAC* commands. The commands will appear automatically in the *Resource (Record)* pane when the operation is executed in a *Model-tool* pane.

3.4.3 Files

There are 11 types of files that are either used or created by *FLAC*. Detailed descriptions of these files are provided in Section 2.10 of the **User's Guide**. The files are, by default, distinguished by their extension. Note that default file names can be changed by the user.

Seven of these file types are important to know before creating and running a *FLAC* model for the first time. These file types are listed and described in Table 3.2.

Table 3.2 *FLAC* File Types

File Type	Default Name	Description
Project files	Any name with the extension ".prj".	Formatted ASCII file containing variables that describe the state of the model and a link to model state (SAV) files associated with the project. Descriptions of user-created plots are also included.
Model state files	FLAC.sav	Binary file created when the **SAVE** command is issued, or [Save] button is pressed. Contains values of all state variables and conditions at the time **SAVE** is invoked.
Data files	Any name or extension can be used --- ".dat" is suggested.	Formatted ASCII file containing a sequence of *FLAC* commands that describe a problem.
Materials files	Any name with the extension ".gmt".	Formatted ASCII file containing the values of material properties selected for application in different projects.
Plot files	Any name can be used for different plot file types.	Several different on-screen and hard-copy plot outputs are provided. Use the [File]/[Print plot] menu item to select output type.
History files	FLAC.his	Formatted ASCII files created when the **HISTORY write** command, or [Utility]/[History]/[Save] tool is issued. History values of selected variables are recorded.
Geometry files	Any name with the extension ".geo".	The file format contains a description of the model geometry. The file is used in the [Sketch] and [Geometry Builder] tools to facilitate creating the model.

3.5 About the Finite Difference Grid

3.5.1 Zones and Gridpoints

The grid is defined by specifying the number of zones, *i*, desired in the horizontal (*x*) direction and the number of zones, *j*, in the vertical (*y*) direction. The grid is thus organized in a row and column fashion. The vertices of the zones meet at gridpoints. Each zone and gridpoint in the grid is uniquely identified by a pair of *i,j* indices. The *i,j* indices of the zones associated with an example grid are shown in Figure 3.3 and for gridpoints in Figure 3.4.

If a Mohr-Coulomb model is assigned to the zones, Mohr-Coulomb properties are stored at the zone centroids and are associated with the zone *i,j* indices.

Note that if there are *p* zones in the *x*-direction and *q* zones in the *y*-direction, then there are *p+1* and *q+1* gridpoints in the *x*- and *y*-directions, respectively.

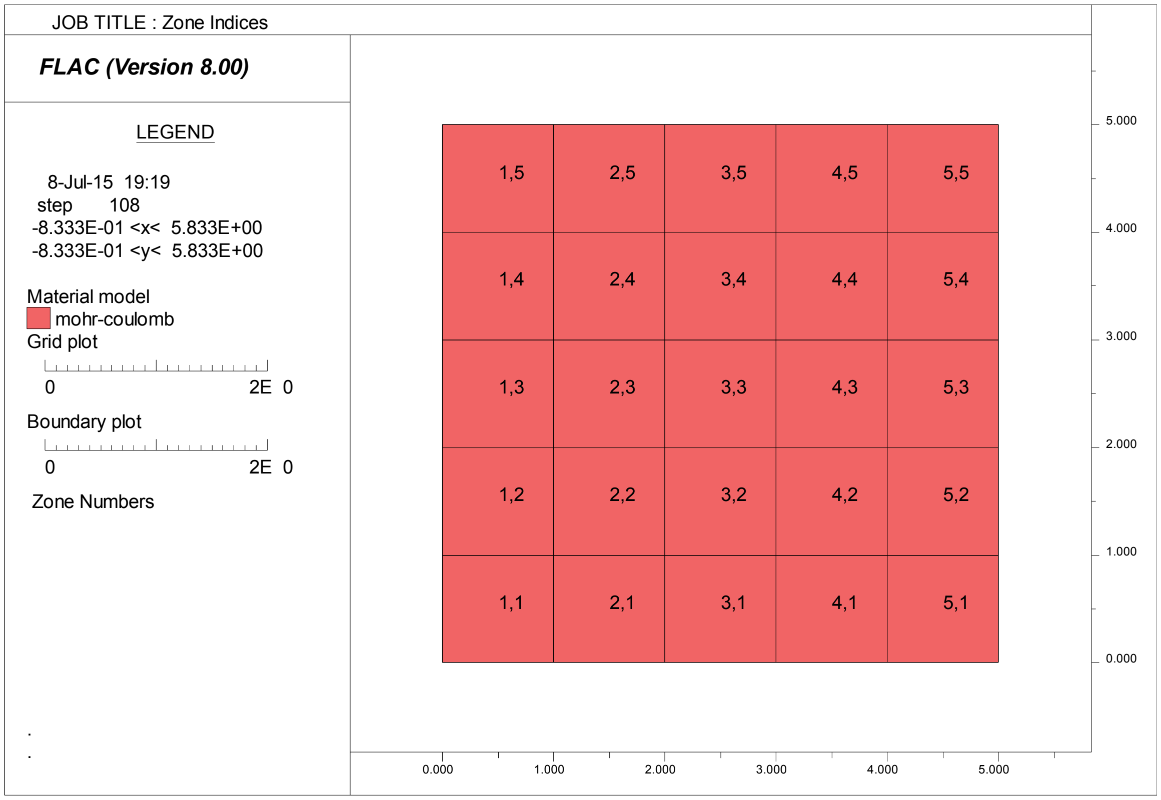


Figure 3.3 Identification of zone *i,j* indices.

Vector quanitites (e.g., forces, velocities, displacements) are stored at gridpoint locations and are associated with gridpoint *i,j* indices.

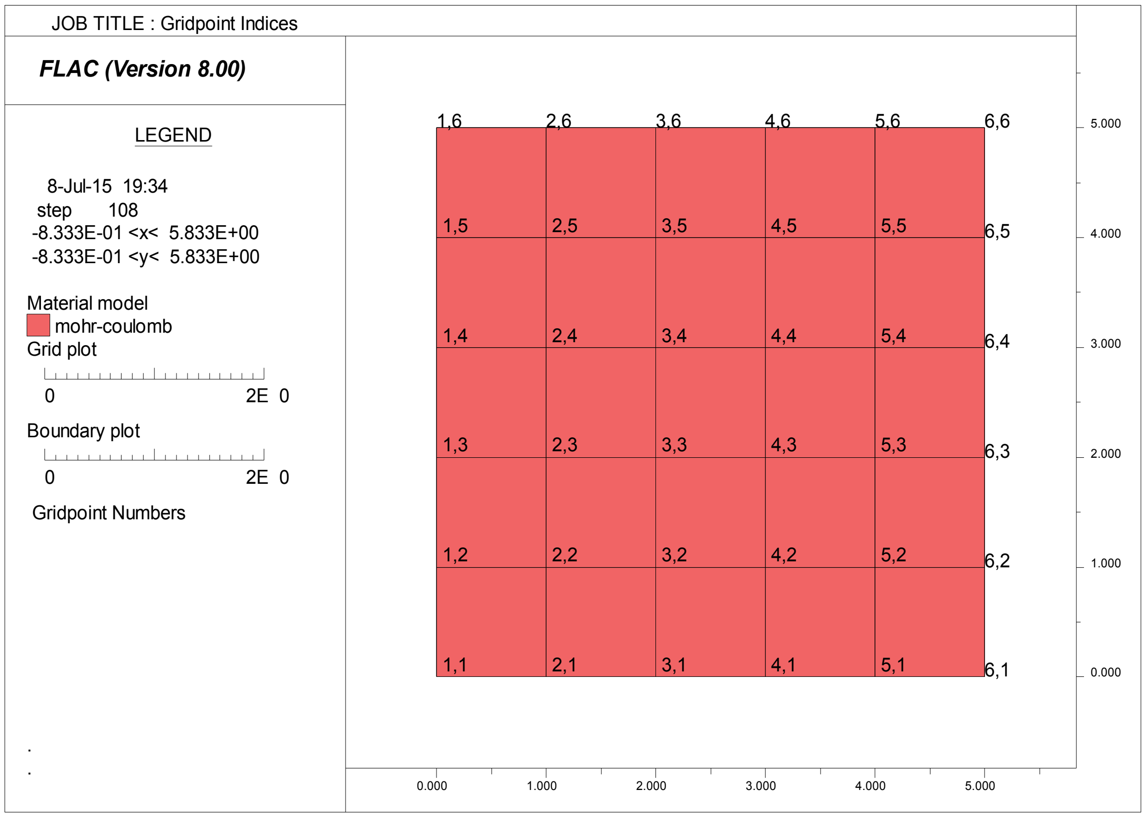


Figure 3.4 Identification of gridpoint *i,j* indices.

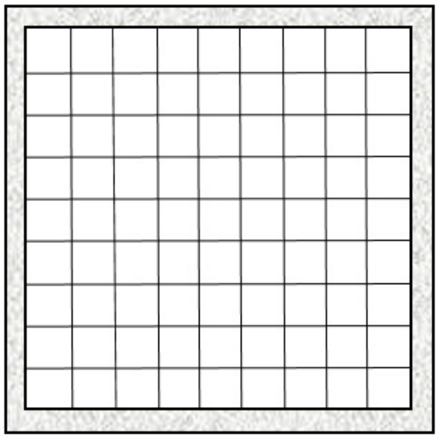
3.5.2 *(i,j)* vs. *(x,y)*

Imagine *x,y* space as a large pin board ruled like graph paper. It has a regular grid but no sense of coordinate values until the origin is positioned somewhere on the grid. Now imagine the *i,j* grid as an elasticated net that is pinned to the *x,y* pin board.

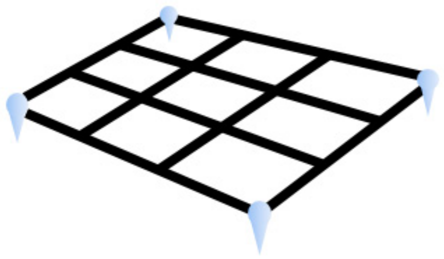
By default, node *i,j (1,1)* is mapped to *x,y (0,0)* and the *x,y* and *i,j* unit lengths are equal. Figure 3.5 illustrates this. The *i,j* nodes can now be moved on the *x,y* space, distorting the mesh in the process, to conform to the model requirements.

Note that the *i,j* identifiers for each node remain unchanged—the *x,y* coordinates mapped to the nodes are recalculated. This holds true for all elements in the model, even for null regions or deformed regions.

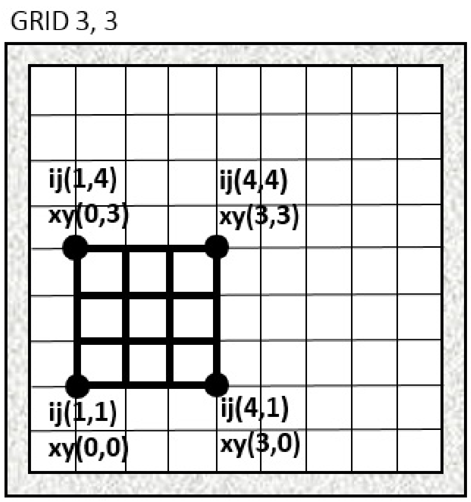
There are many more tools available for building a mesh to suit model geometries and many ways to use them. Some of these will be explored as meshes are developed in the example problems that follow.



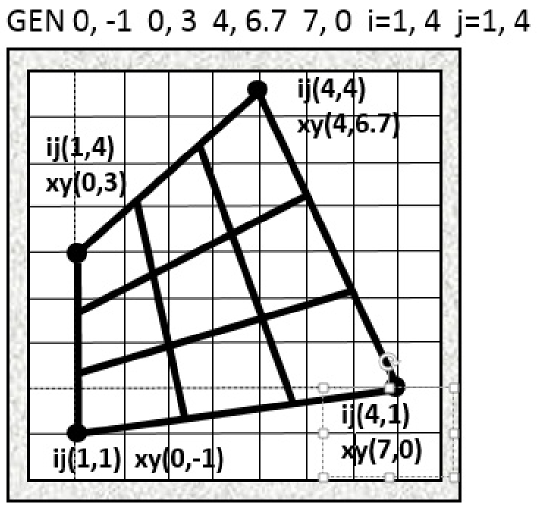
This is the gridded pinboard representing **xy** space. At this stage the origin **xy(0,0)** has not been positioned.



The elasticated net represents **ij** space. It has pins at as many or as few nodes as required to outline the expected pattern.



*The mesh representing **ij** space is pinned onto the **xy** space. By default, the unit length is equivalent in both spaces and **xy(0,0)** matches **ij(1,1).***



*The pins are then moved to suit the model geometry. All the nodes on the mesh retain their original **ij** values although they are now mapped to different **xy** values.*

Figure 3.5 *(i,j)* **vs.** *(x,y)* **space.**

3.6 Some Notes on *FISH*

A guide to using *FISH* can be found in Sections 2 and 3 of the ***FISH* in FLAC** volume.

FISH is a programming language embedded within *FLAC* that enables the user to define new variables and functions. These functions may be used to extend *FLAC*'s usefulness or implement new constitutive models. For example, new variables may be plotted or printed, special grid generators may be implemented, servo control may be applied to a numerical test, unusual distributions of properties may be specified, and parameter studies may be automated. *FISH* is a compiler: programs entered via a *FLAC* data file are translated into a list of instructions stored in *FLAC*'s memory space. The original source program is not retained by *FLAC*. Whenever a *FISH* function is invoked, its compiled code is executed.

The compiled form of a *FISH* function is stored in *FLAC*'s memory space. The **SAVE** command saves this function and the current values of associated variables.

FISH functions are simply embedded in a normal *FLAC* data file. Lines following the command **DEF** are processed as belonging to the function named on the **DEF** line. The function terminates when the command **END** is encountered. The function is only executed when its name is specified as a command.

An important construct in *FISH* is the sequence

```
COMMAND
   ; (FLAC commands go here)
ENDCOMMAND
```

Here, *FLAC* commands are given from within the *FISH* function. This enables us, among other things, to control an entire *FLAC* run—for example, to do a parametric study or to capture a series of screen plots to a movie. Most valid *FLAC* commands can be embedded between these statements. These are demonstrated later in Chapters 4 and 5.

3.6.1 Naming Conventions

The *FLAC* manuals contain detailed discussions on the use of *FISH*. Tables of reserved words are provided and should be consulted to avoid conflicts.

FISH variables, function names and statements must be spelled out in full. No continuation lines are allowed; intermediate variables may be used to split complex expressions. *FISH* is case insensitive, with all names converted to lower case. Spaces are significant and serve to separate variables, keywords, and values. No embedded blanks are allowed in function names, but extra whitespace may be used to aid readability. Any characters following a semicolon (;) are ignored. Variable or function names must start with a non-numeric character and must not contain any of the following symbols:

```
.   ,   *  /  +   -   ^   =   <   >   #   (   )   [   ]   @   ;
```

User-defined names can be any length, although they may be truncated in print-out and plot captions due to line length limitations.

3.6.2 Scope of Variables

Variable and function names are recognized globally (except for property variables associated with user-defined constitutive models). As soon as a name is mentioned in a valid *FISH* program line, it is thereafter recognized globally, both in *FISH* code and in *FLAC* commands. It also appears in the list of variables displayed when the **PRINT FISH** command is given. A variable may be given a value in one *FISH* function and used in another function or in a *FLAC* command; the value is retained until it is changed. The values of all variables are saved when the *FLAC* **SAVE** command is given and restored by the **RESTORE** command.

3.6.3 Assignment of Data Types

There are three data types used for *FISH* variables or function values.

- Integer: Exact numbers in the range -2,147,483,648 to +2,147,483,647.
- Floating point: Real values with about 6 decimal digits of precision, with a range of approximately 10^{-300} to 10^{300}.
- String: A packed sequence of printable characters enclosed by single quotes.

A numeric variable can change type dynamically, adjusting according to context. A variable's type can be pre-assigned, but this is only necessary in constitutive models optimized for more efficient calculation.

3.6.4 A Simple *FISH* Function

Whenever a number is expected in the *FLAC* input line, the name of a *FISH* variable or function may be substituted. This is a very powerful feature of *FLAC* because it allows parameter changes without the need to change many numbers in the data file.

For example, assume that the Young's Modulus (E) and Poisson's Ratio (v) of a material are known. Because the *FLAC* stress-strain calculation requires the bulk (K) and shear (G) moduli, these may be calculated using the formulae:

$$G = E / 2(1 + v)$$

and

$$K = E / 3(1 - 2 v)$$

The code for the *FISH* function **Derive*KG*** using these equations is shown in Example 3.1.

Example 3.1 *FISH* function to calculate *K* and *G* from *E* and *v*.

```
================================================================
; -----------------------------------------------------
;  Function to calculate K and G from E and nu
; -----------------------------------------------------
def  DeriveKG
     s_mod = e_mod / (2.0 * (1.0 + p_ratio))
     b_mod = e_mod / (3.0 * (1.0 – 2.0 * p_ratio))
end
================================================================
```

The *FISH* parameters **p_ratio** and **e_mod** are specified with the **SET** command before the function is executed.

SET e_mod = 5e8 p_ratio = 0.25

The function is executed by giving the function name:

DeriveKG

The values are now computed for the bulk and shear moduli and assigned to the *FISH* variables **b_mod** and **s_mod**, respectively.

The contents of these variables are then assigned to the *FLAC* grid using the **PROPERTY** command:

```
PROPERTY dens = 1000 bulk = b_mod shear = s_mod
```

The validity of this approach may be checked by printing out bulk and shear moduli in the usual way (i.e., **PRINT bulk** and **PRINT shear**).

Chapter 4
A Simple Analysis: Stability of a Silty Clay Slope

4.1 Background

This example illustrates the *FLAC* modeling procedure and how it can be applied to parametric studies. The example first introduces primary modeling commands and then shows how *FISH* is used to control a parametric analysis.

Chen (2007)[1] provides an analytical solution for the factor of safety of a homogeneous embankment in silty clay. The factor is calculated based upon limit analysis theory, for a 10-m high embankment with a slope angle of 45°. The unit weight of the silty clay is 20 kN/m^3, the cohesion is 12.38 kPa, and the friction angle is 20°. An applied gravitational loading is specified with a magnitude of 10 m/sec^2. For these conditions, the factor of safety (FoS) is calculated to be exactly 1.0.

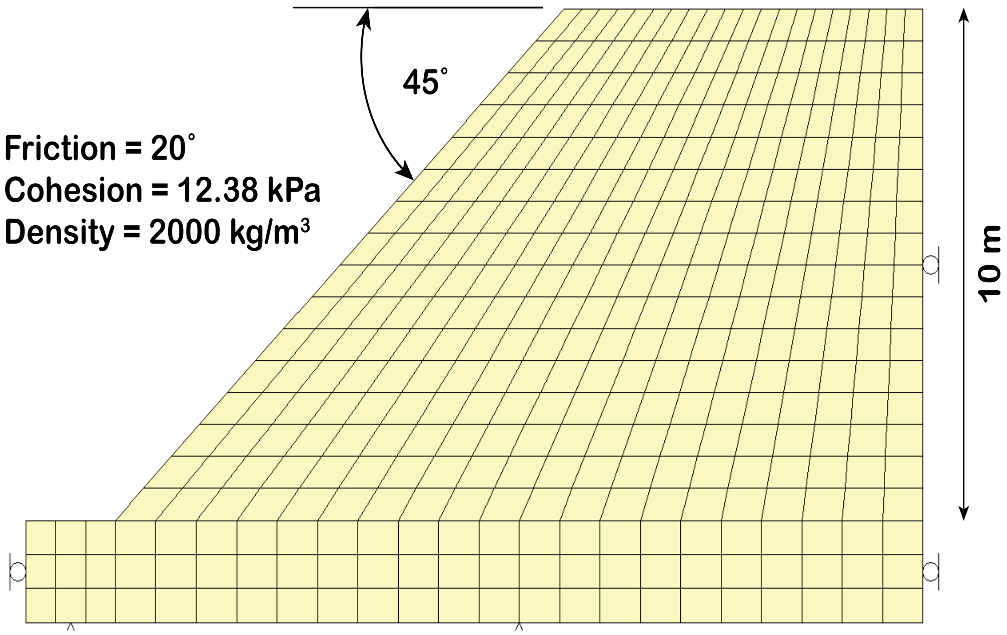


Figure 4.1 Slope stability problem, solved analytically by Chen (2007), has a factor of safety of exactly 1.0 for a slope angle of 45°.

Two questions are asked in this *FLAC* exercise:

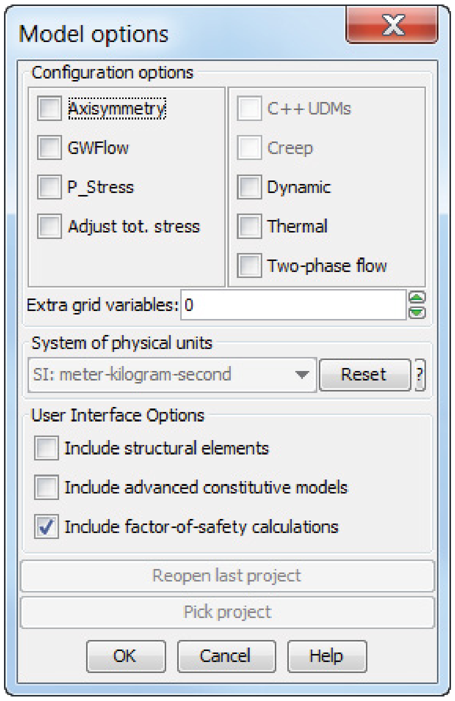
1. How well does the *FLAC* calculation for factor of safety compare with the Chen (2007) solution?
2. What is the slope angle that will provide a factor of safety of 1.3 for this silty clay embankment?

1. Chen, W.-F. ***Limit Analysis and Soil Plasticity***. J. Ross Publishing (2007).

4.2 Solution Approach

Unlike conventional slope stability methods that examine a discrete pattern of potential failure planes, which may or may not identify the most probable worst case, *FLAC* will find the failure mechanism and identify the failure plane directly by reducing the strength of the material until failure occurs. The critical strength will be compared to the actual strength to determine the factor of safety. This method is called the **strength reduction method** to calculate a safety factor.

FLAC includes built-in logic based on the strength reduction method to calculate a factor of safety automatically. The method is described in the **Factor of Safety** volume of the documentation set. The factor of safety calculation is executed when the **SOLVE fos** command is issued and can be accessed by selecting “Include factor of safety calculations” in the *Model options* dialog when starting this *FLAC* project. See Figure 4.2.


Model options

Configuration options

[ ] Axisymmetry
[ ] GWFlow
[ ] P_Stress
[ ] Adjust tot. stress
[ ] C++ UDMs
[ ] Creep
[ ] Dynamic
[ ] Thermal
[ ] Two-phase flow

Extra grid variables: 0

System of physical units

SI: meter-kilogram-second

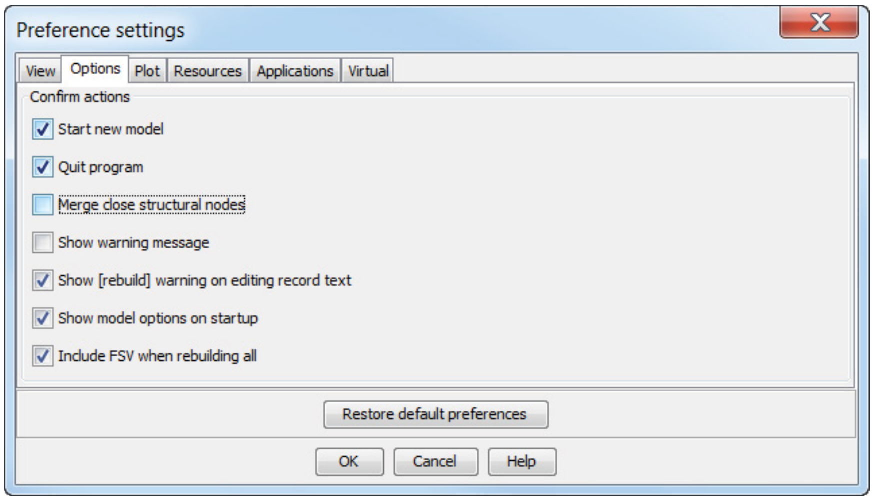
User Interface Options

[ ] Include structural elements
[ ] Include advanced constitutive models
[✓] Include factor-of-safety calculations

Figure 4.2 Select “Include factor-of-safety calculations” to access the built-in factor of safety logic in *FLAC*.

These two questions are addressed in one *FLAC* model project divided into two stages. First, a model is created to compare the *FLAC* calculation for a factor of safety to the analytical solution (FoS = 1.0). Then, the model is rerun several times with different slope angles until a slope angle is found that provides a FoS of 1.3. The second stage uses a *FISH* function to both adjust the slope angle and repeat the calculation automatically until the slope angle is determined that results in the required factor of safety.

PLEASE NOTE: In this exercise, multiple factor of safety calculations will be performed in one run. By default, *FLAC* issues a message and pauses execution when an FoS calculation is completed. This message should be disabled when multiple FoS calculations are performed in one run. Before beginning the exercise, open the [File]/[Preference Settings] menu item and uncheck “Show warning message” as shown in Figure 4.3. This will permit multiple FoS calculations without interruption.



Confirm actions

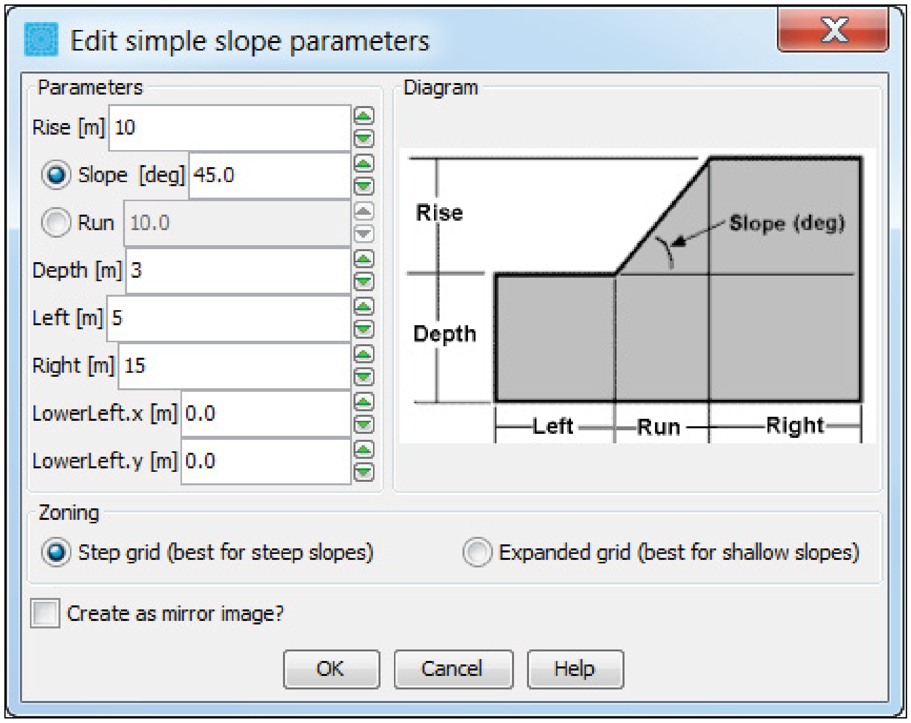
- ✓ Start new model
- ✓ Quit program
- ☐ Merge close structural nodes
- ☐ Show warning message
- ✓ Show [rebuild] warning on editing record text
- ✓ Show model options on startup
- ✓ Include FSV when rebuilding all

Figure 4.3 Preference settings window.

4.3 Model Description

A project file, “slope.prj”, is created for this example in the *Project file* dialog and saved in a selected directory (e.g., “C:\itasca\flac800\datafiles\flacbasics”).

This model is easily created in *FLAC* by using the [Build]/[Generate]/[Slope] tool, as shown in Figure 4.4. The slope parameters are entered as shown in the figure. The right side of the slope is located 15 m away from the slope crest to ensure that any failure surface that develops will not extend to a model boundary. A “step-grid” is selected because “steep slope” grids (i.e., grids with slope angles greater than 30°) are expected to be created for this exercise.



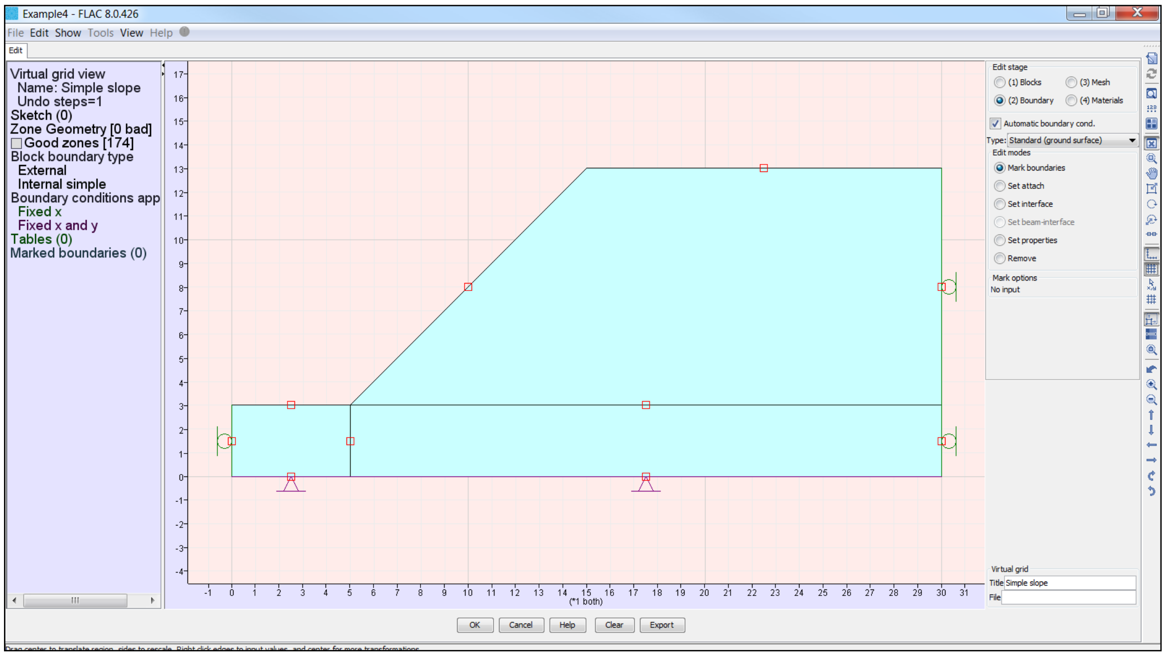
Rise [m] 10
[✓ Slope [deg]] 45.0
Run 10.0
Depth [m] 3
Left [m] 5
Right [m] 15
LowerLeft.x [m] 0.0
LowerLeft.y [m] 0.0

Zoning
[✓ Step grid (best for steep slopes)] Expanded grid (best for shallow slopes)

☐ Create as mirror image?

Figure 4.4 Enter simple slope parameters using the [Build]/[Generate]/[Slope] tool.

Boundary conditions, zoning, and material properties are now assigned using the [Build]/[Virtual]/[Edit] tool. Standard boundary conditions (i.e., roller boundaries on the sides of the model and fixed base) are prescribed by checking “Automatic boundary cond.” in the [Boundary] *Edit stage*; see Figure 4.5.

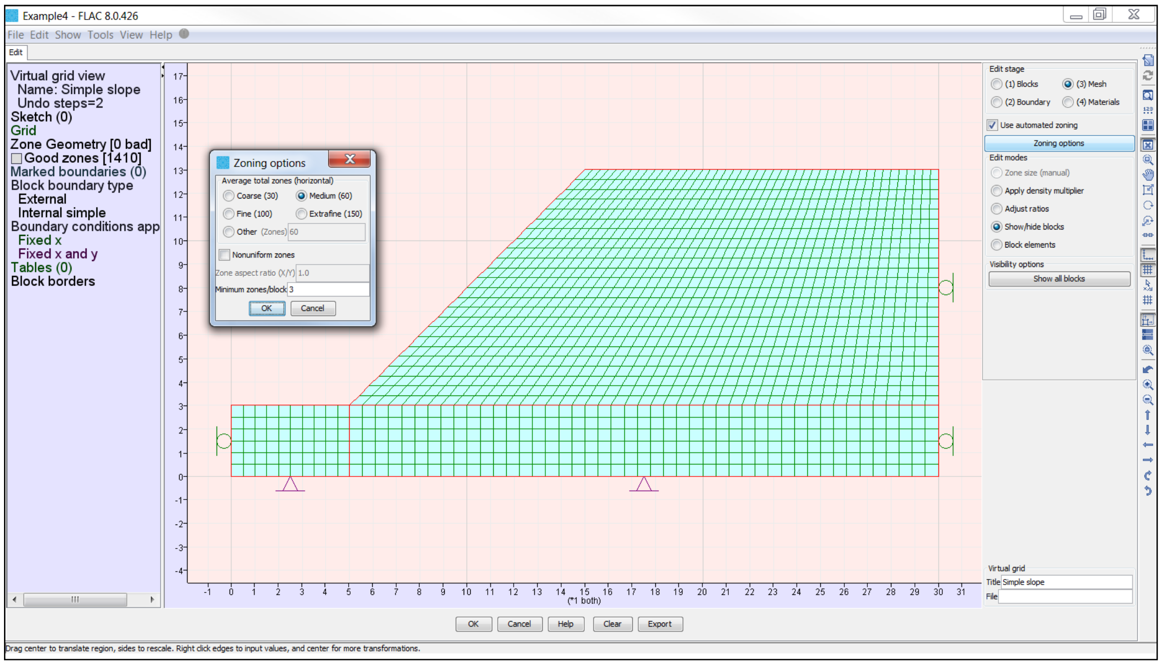


Virtual grid view
Name: Simple slope
Undo steps=1
Sketch (0)
Zone Geometry [0 bad]
Good zones [174]
Block boundary type
External
Internal simple
Boundary conditions app
Fixed x
Fixed x and y
Tables (0)
Marked boundaries (0)

Edit stage: (1) Blocks [✓ (2) Boundary] (3) Mesh (4) Materials
[✓ Automatic boundary cond.]
Type: Standard (ground surface)
Edit modes: [✓ Mark boundaries] Set attach, Set interface, Set beam-interface, Set properties, Remove
Mark options: No input
Virtual grid: Title Simple slope; File

Figure 4.5 Assign automatic boundary conditions.

The model is now "zoned" (i.e., divided into a mesh of quadrilateral-shaped zones) using the [Mesh] *Edit stage*. For simple model shapes, zoning is best accomplished by checking [Use automated zoning] and selecting [Zoning options]. A *Zoning options* dialog opens. In most cases, fast solutions with reasonable accuracy can be obtained by selecting a "Medium" zoned mesh. In this example, a "Medium" mesh is selected, as shown in Figure 4.6. Finer zoning will provide a more accurate *FLAC* calculation for comparison to the Chen solution.



Virtual grid view
Name: Simple slope
Undo steps=2
Sketch (0)
Grid
Zone Geometry [0 bad]
Good zones [1410]
Marked boundaries (0)
Block boundary type
External
Internal simple
Boundary conditions app
Fixed x
Fixed x and y
Tables (0)
Block borders

Zoning options
Average total zones (horizontal): Coarse (30) [✓ Medium (60)] Fine (100) Extrafine (150) Other (Zones) 60
Nonuniform zones
Zone aspect ratio (X/Y) 1.0
Minimum zones/block 3

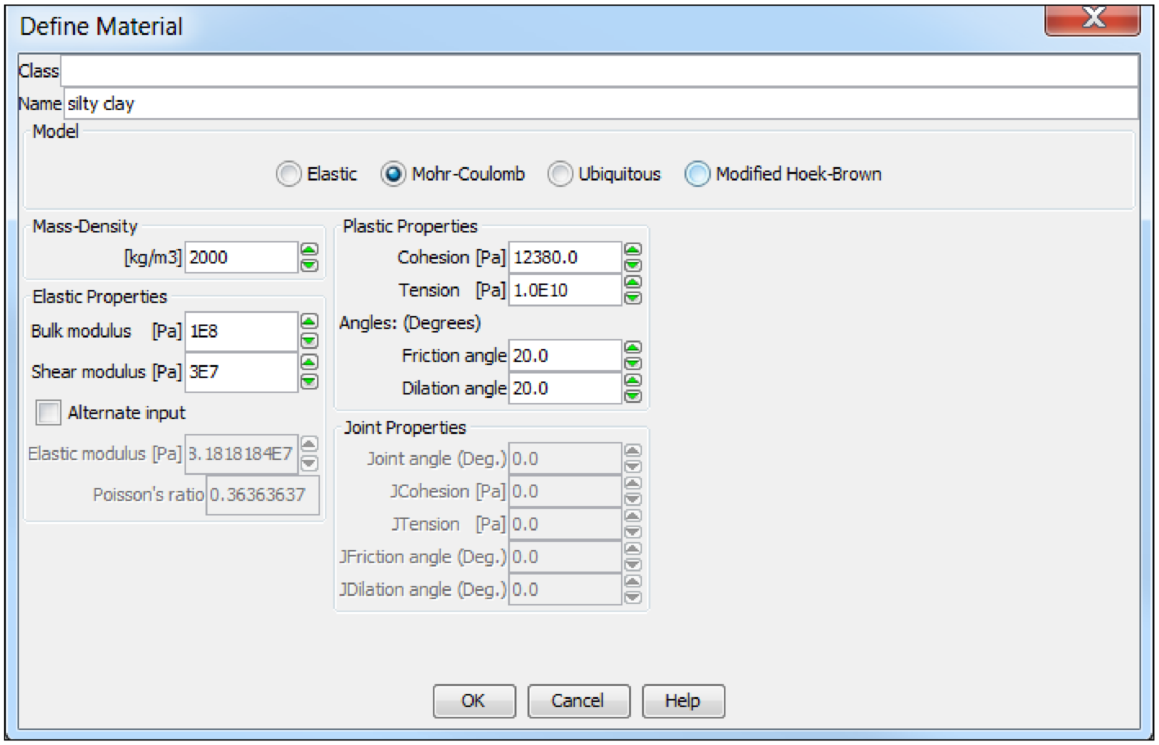
Edit stage: (1) Blocks (2) Boundary [✓ (3) Mesh] (4) Materials
[✓ Use automated zoning]
Zoning options
Edit modes: Zone size (manual), Apply density multiplier, Adjust ratios [✓ Show/hide blocks] Block elements
Visibility options: Show all blocks
Virtual grid: Title Simple slope; File

Drag center to translate region, sides to rescale. Right click edges to input values, and center for more transformations.

Figure 4.6 Prescribe a "Medium" zoned mesh.

A Mohr-Coulomb constitutive model and properties are assigned to the zones using the [Materials] *Edit stage*. Press the [Create] button to open a *Define Material* dialog, as shown in Figure 4.7. A material group name, *silty clay*, is assigned to the material. Then, the Mohr-Coulomb material model is selected and properties are entered as shown in the figure. Note that mass density is input by dividing the unit weight by the gravitational magnitude. The choice of elastic properties is arbitrary because these properties do not significantly affect the solution in this particular case. Elastic moduli can be assigned as either bulk and shear moduli, or elastic (Young's) modulus and Poisson's ratio in the *Define Material* dialog. The dilation angle is set equal to the friction angle because the analytical solution given by Chen (2007) assumes associated plastic flow in the limit analysis. The tensile strength is set to a high value because tensile failure is not considered in the Chen solution.

The Mohr-Coulomb model and properties are automatically assigned to all zones in the model by pressing [Set all] in the [Materials] *Edit stage*, as shown in Figure 4.8.


Define Material

Class:
Name: silty clay

Model: ( ) Elastic [✓ Mohr-Coulomb] ( ) Ubiquitous ( ) Modified Hoek-Brown

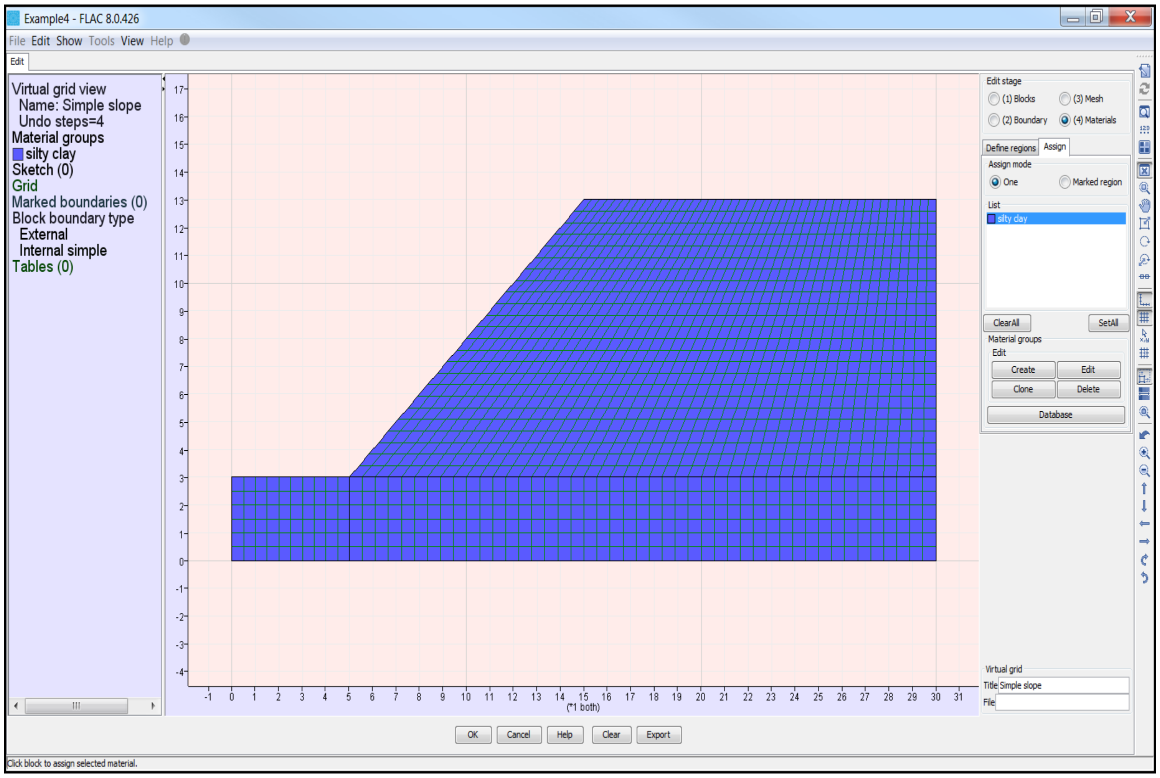
Mass-Density
[kg/m3] 2000

Elastic Properties
Bulk modulus [Pa] 1E8
Shear modulus [Pa] 3E7
[ ] Alternate input
Elastic modulus [Pa] 8.1818184E7
Poisson's ratio 0.36363637

Plastic Properties
Cohesion [Pa] 12380.0
Tension [Pa] 1.0E10
Angles: (Degrees)
Friction angle 20.0
Dilation angle 20.0

Joint Properties
Joint angle (Deg.) 0.0
JCohesion [Pa] 0.0
JTension [Pa] 0.0
JFriction angle (Deg.) 0.0
JDilation angle (Deg.) 0.0

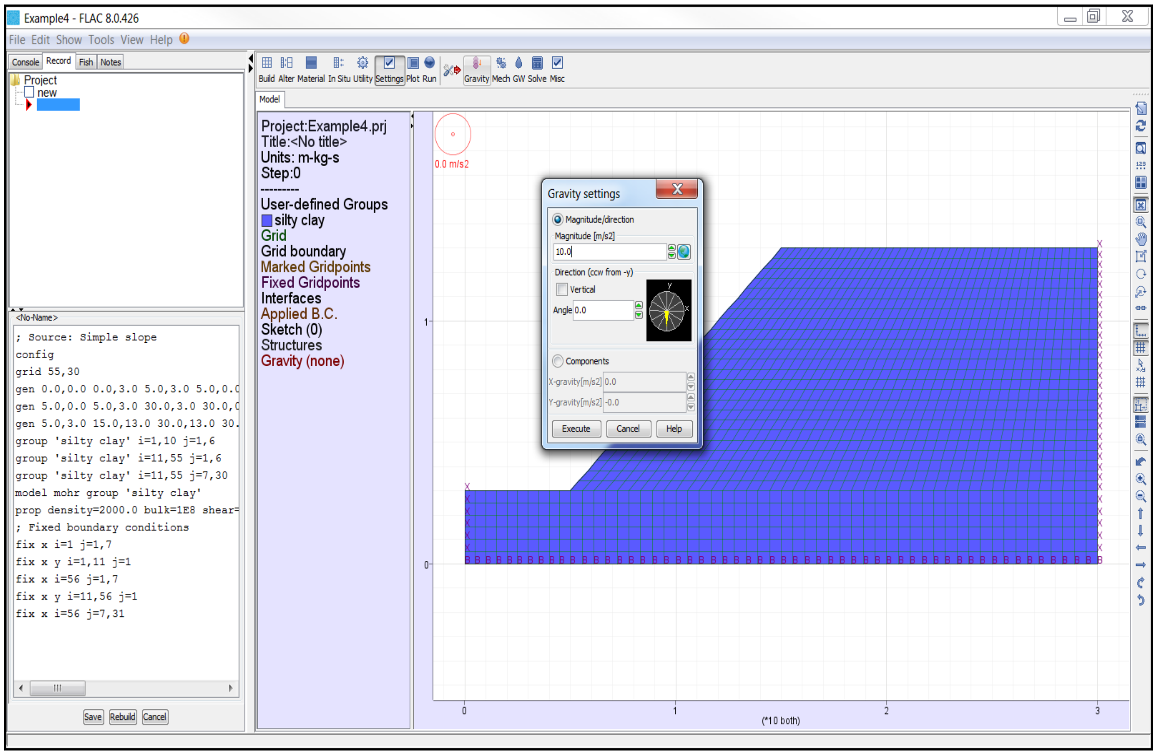
Figure 4.7 Define a Mohr-Coulomb material and properties.



Virtual grid view
Name: Simple slope
Undo steps=4
Material groups
🟦 silty clay
Sketch (0)
Grid
Marked boundaries (0)
Block boundary type
External
Internal simple
Tables (0)

Figure 4.8 Assign the material to all zones in the model by pressing [Set all].

The created "virtual" model is now ready to be transformed into a *FLAC* model defined by a sequence of *FLAC* commands. This is accomplished by pressing the [Build]/[Virtual]/[Execute] button. Gravitational loading of 10 m/sec^2 is then applied using the [Settings]/[Gravity] tool, as shown in Figure 4.9.


Example4 - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Project
new
<No-Name>
Save
Rebuild
Cancel
Build Alter Material In Situ Utility Settings Plot Run
Gravity Mech GW Solve Misc
Model
0.0 m/s2
(*10 both)

```
; Source: Simple slope
config
grid 55,30
gen 0.0,0.0 0.0,3.0 5.0,3.0 5.0,0.
gen 5.0,0.0 5.0,3.0 30.0,3.0 30.0,
gen 5.0,3.0 15.0,13.0 30.0,13.0 30.
group 'silty clay' i=1,10 j=1,6
group 'silty clay' i=11,55 j=1,6
group 'silty clay' i=11,55 j=7,30
model mohr group 'silty clay'
prop density=2000.0 bulk=1E8 shear=
; Fixed boundary conditions
fix x i=1 j=1,7
fix x y i=1,11 j=1
fix x i=56 j=1,7
fix x y i=11,56 j=1
fix x i=56 j=7,31
```

Project:Example4.prj
Title:<No title>
Units: m-kg-s
Step:0

User-defined Groups
🟦 silty clay
Grid
Grid boundary
Marked Gridpoints
Fixed Gridpoints
Interfaces
Applied B.C.
Sketch (0)
Structures
Gravity (none)

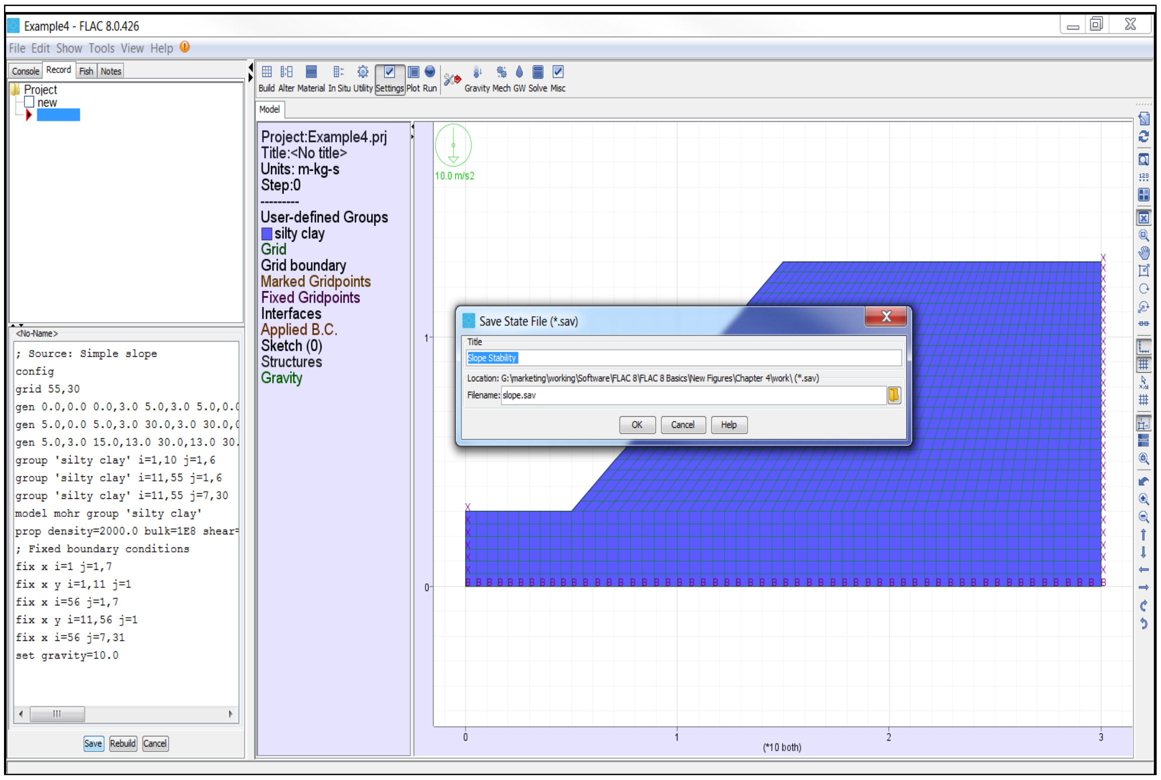
Gravity settings
[✓ Magnitude/direction]
Magnitude [m/s2] 10.0
Direction (ccw from -y)
Vertical
Angle 0.0
Components
X-gravity[m/s2] 0.0
Y-gravity[m/s2] -0.0
Execute Cancel Help

Figure 4.9 Apply gravitational acceleration.

Before performing the factor of safety calculation, the model state must be saved by pressing the [Save] button at the bottom of the *Record* pane. The *FLAC* commands created for this model are shown in the *Record* pane in Figure 4.10 and listed below in Example 4.1. All of the commands are described in Section 1 of the **Command Reference**.

Example 4.1 *FLAC* commands to create 45° slope model.

```
==========================================================
grid 55,30
gen 0.0,0.0 0.0,3.0 5.0,3.0 5.0,0.0 i=1,11 j=1,7
gen 5.0,0.0 5.0,3.0 30.0,3.0 30.0,0.0 i=11,56 j=1,7
gen 5.0,3.0 15.0,13.0 30.0,13.0 30.0,3.0 i=11,56 j=7,31
group 'silty clay' i=1,10 j=1,6
group 'silty clay' i=11,55 j=1,6
group 'silty clay' i=11,55 j=7,30
model mohr group 'silty clay'
prop density=2000.0 bulk=1E8 shear=3E7 group 'silty clay'
prop cohesion=12380.0 friction=20.0 group 'silty clay'
prop dilation=20.0 tension=1.0e10 group 'silty clay'
fix x i=1 j=1,7
fix x y i=1,11 j=1
fix x i=56 j=1,7
fix x y i=11,56 j=1
 fix x i=56 j=7,31
 set gravity=10.0
==========================================================
```


Example4 - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Project
new
Build Alter Material In Situ Utility Settings Plot Run
Gravity Mech GW Solve Misc
Model
10.0 m/s2
(*10 both)
Save Rebuild Cancel

Project:Example4.prj
Title:<No title>
Units: m-kg-s
Step:0
User-defined Groups
silty clay
Grid
Grid boundary
Marked Gridpoints
Fixed Gridpoints
Interfaces
Applied B.C.
Sketch (0)
Structures
Gravity

<No-Name>

```
; Source: Simple slope
config
grid 55,30
gen 0.0,0.0 0.0,3.0 5.0,3.0 5.0,0.0
gen 5.0,0.0 5.0,3.0 30.0,3.0 30.0,0
gen 5.0,3.0 15.0,13.0 30.0,13.0 30.
group 'silty clay' i=1,10 j=1,6
group 'silty clay' i=11,55 j=1,6
group 'silty clay' i=11,55 j=7,30
model mohr group 'silty clay'
prop density=2000.0 bulk=1E8 shear=
; Fixed boundary conditions
fix x i=1 j=1,7
fix x y i=1,11 j=1
fix x i=56 j=1,7
fix x y i=11,56 j=1
fix x i=56 j=7,31
set gravity=10.0
```

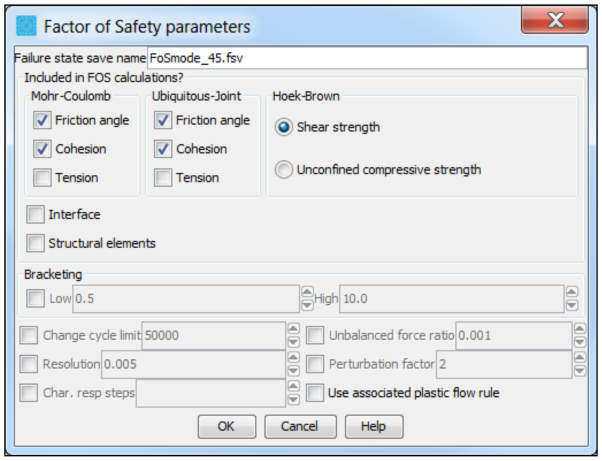
Save State File (*.sav)
Title: Slope Stability
Location: G:\marketing\working\Software\FLAC 8\FLAC 8 Basics\New Figures\Chapter 4\work\ (*.sav)
Filename: slope.sav
OK Cancel Help

Figure 4.10 The model state is saved before performing the factor of safety calculation.

4.3.1 Stage 1: Calculate FoS for 45° Slope

The extension ".fsv" distinguishes the state as the "failure state" for the model.

The factor of safety calculation is run by selecting [Run]/[SolveFoS]. This opens a *Factor of Safety parameters* dialog, as shown in Figure 4.11. A failure-state SAV file named "FoSmode_45.fsv" is assigned, and the default strength factors, cohesion, and friction angle for the Mohr-Coulomb material are selected for strength reduction. When [OK] is pressed, the **SOLVE fos** command is executed and the calculation begins.



Factor of Safety parameters

Failure state save name: FoSmode_45.fsv

Included in FOS calculations?

Mohr-Coulomb
- ✓ Friction angle
- ✓ Cohesion
- ☐ Tension

Ubiquitous-Joint
- ✓ Friction angle
- ✓ Cohesion
- ☐ Tension

Hoek-Brown
- ● Shear strength
- ○ Unconfined compressive strength

☐ Interface

☐ Structural elements

Bracketing
- ☐ Low 0.5 High 10.0

☐ Change cycle limit 50000 ☐ Unbalanced force ratio 0.001

☐ Resolution 0.005 ☐ Perturbation factor 2

☐ Char. resp steps ☐ Use associated plastic flow rule

Figure 4.11 Perform a factor of safety calculation.

When the run is complete, the factor of safety result can be viewed by selecting the tool [Plot]/[Model] and then select the *Factor of Safety* plot item. A factor of 1.01 is calculated for this model, and the failure surface, identified by shear strain rate contours and velocity vectors, is plotted as shown in Figure 4.12. This result is considered acceptable for this exercise. The factor approaches 1.0 as the zoning becomes finer. For example, a factor of 1.0 is calculated if the [Extra fine] zone size is selected. See Figure 4.13.

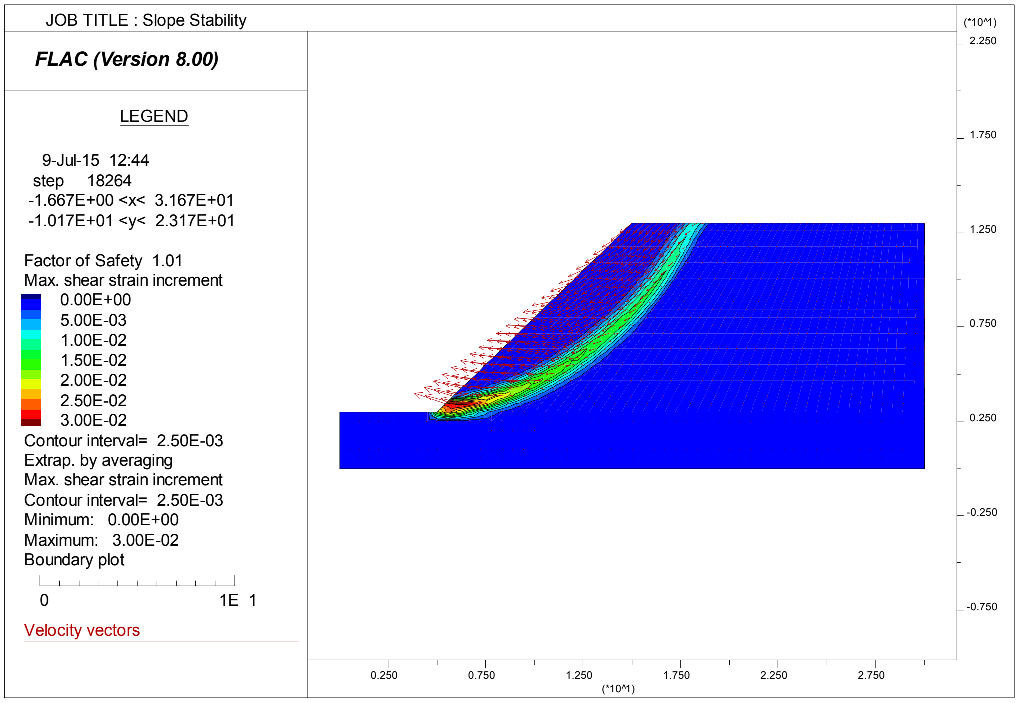


Figure 4.12 Factor of safety result for [Medium] zoning (FoS = 1.01).

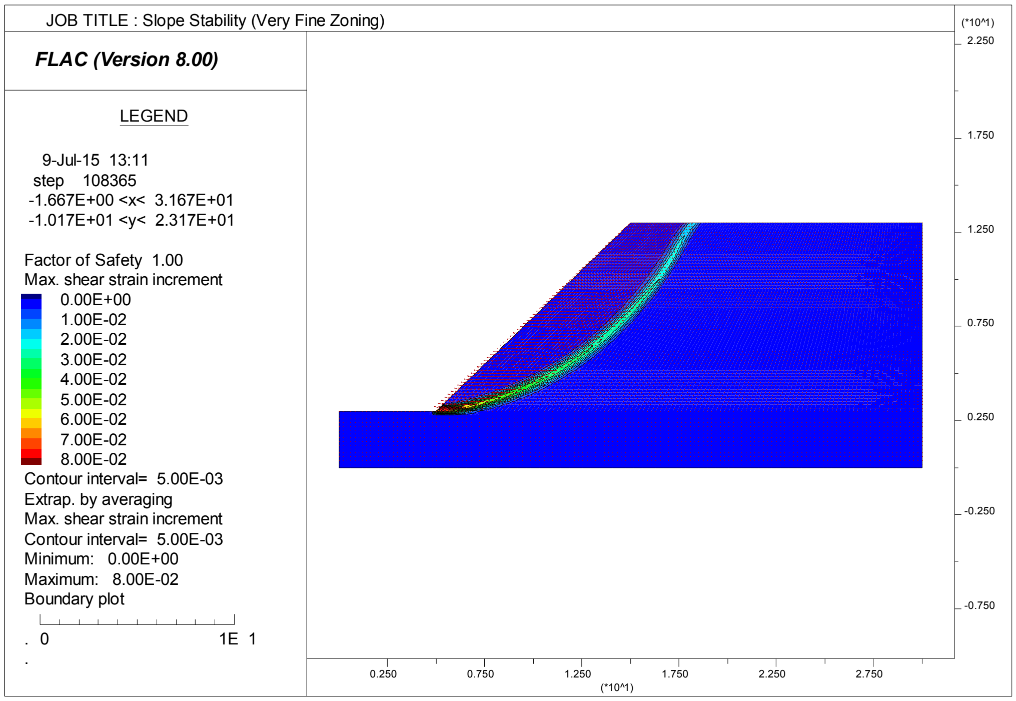


Figure 4.13 Factor of safety result for [Extra fine] zoning (FoS = 1.00).

4.3.2 Stage 2: Find Slope Angle for FoS = 1.3

This *FLAC* model will now be used to begin a series of factor of safety calculations for models with different slope angles to determine the angle that results in a safety factor of 1.3. The *FLAC* model commands, shown in Example 4.1, are embedded within a *FISH* function that controls the slope angle and the factor of safety calculation. The function, named **slope_angle**, is listed in Example 4.2. This *FISH* function is created for the *FLAC* model using the *Fish editor* pane.

The **slope_angle** *FISH* function uses a binary (bracketing) search algorithm to find the desired slope angle in a low number of model runs. Upper and lower bracketing slope angles are input to start the calculation. These angles are defined by the *x*-coordinate of the slope crest. **LXtop** is the *FISH* parameter name for the left bracket *x*-coordinate, and **RXtop** is the right bracket *x*-coordinate. **LXtop** is set to 15.0, which is the *x*-coordinate location of the slope crest for the 45° slope. **RXtop** is set to 30.0, which is the *x*-coordinate of the right model boundary.

It is assumed that the *x*-coordinate that corresponds to the slope angle for FoS = 1.3 will fall between **LXtop** and **RXtop**. If a solution for FoS = 1.3 within this bracket range cannot be reached, then the right boundary of the model (and **Rxtop**) will need to be moved farther away from the slope crest.

In addition to **LXtop** and **RXtop**, the required factor of safety (FoS = 1.3) is also input. This parameter is named **fos_limit** in the *FISH* function.

Example 4.2 ***FISH*** **function controlling the model.**

```
=========================================================================
def slope_angle
 fos_limitp = fos_limit + 0.005
 fos_limitm = fos_limit - 0.005
 Xtop = (LXtop + RXtop)/2.0
 loop nn (1,10)
   command
;;;;;;;;;;;;;;;;;;;;;
;;; add FLAC model commands  🡐 insert the initial 45°  slope FLAC model commands
;;;                            between the command and endcommand FISH statements
;;;;;;;;;;;;;;;;;;;;;
   endcommand
   if fos < fos_limit then
    LXtop = Xtop
   Xtop = (Xtop + RXtop)/2.0
   else
    RXtop = Xtop
    Xtop = (Xtop + LXtop) / 2.0
   endif
   if fos < fos_limitp then
    if fos > fos_limitm then
     x_slope = Xtop - 5.0
     y_slope = 10.0
     sl_angle = atan2(y_slope,x_slope)/degrad
     oo = out(‘ Slope angle = ‘ + string(sl_angle)+’ degrees’)
     exit
    endif
   endif
 endloop
end
=========================================================================
```

The procedure to create and execute the *FISH* function to find the slope angle that results in **fos_limit = 1.3** is as follows.

1. The *FLAC* commands, listed in Example 4.1, are embedded in the **slope_angle** *FISH* function between the **command** and **endcommand** *FISH* statements, as shown in Example 4.2.
2. Modifications are made to the embedded *FLAC* commands as shown in Example 4.3:
    a. The **GRID 55,30** command is replaced with the command **MODEL null**. The grid will be created outside the *FISH* function. The **MODEL null** command initializes model parameters before each FoS calculation.
    b. The *x*-coordinate of the slope crest (15.0) is replaced with the *FISH* variable named **Xtop**. In this way, by simply changing **Xtop**, a model with a different slope angle can be created.
    c. The dilation angle and tensile strength of the silty clay material are set to zero. These are more realistic values for soft soils.
    d. The **SOLVE fos no_restore file slope_fos_1_3.fsv** command is added at the end of the *FLAC* commands. As a result, the FoS calculation will be performed for each slope model, and the failure state (“slope_fos_1_3.fsv”) will be shown when the entire calculation is complete. (If the keyword phrase **no_restore** is not given, then the original save state is restored when the calculation ends.)

3. The bracketing search is limited to a maximum of 10 iterations with the use of the **loop nn(1,10)** statement. Use the command **PRINT nn** when the run is complete to find the number of iterations that are actually performed.
4. For each FoS iteration, a new **Xtop** is calculated as the average of **LXtop** and **RXtop**.
5. After each FoS calculation, the resulting factor is compared to **fos_limit**. The *FISH* function checks if the calculated value is within a tolerance (by default set to 0.005) of **fos_limit**.
6. If the calculated factor is not within the tolerance, a new **Xtop** is calculated within a reduced bracket range, based upon the location of the calculated factor of safety to **fos_limit**. A new model is created and a new FoS calculation is made.
7. Steps 4, 5, and 6 are repeated until the calculated factor of safety is within the tolerance of **fos_limit**.

Example 4.3 *FLAC* commands embedded in "slope_angle.fis" FISH function.

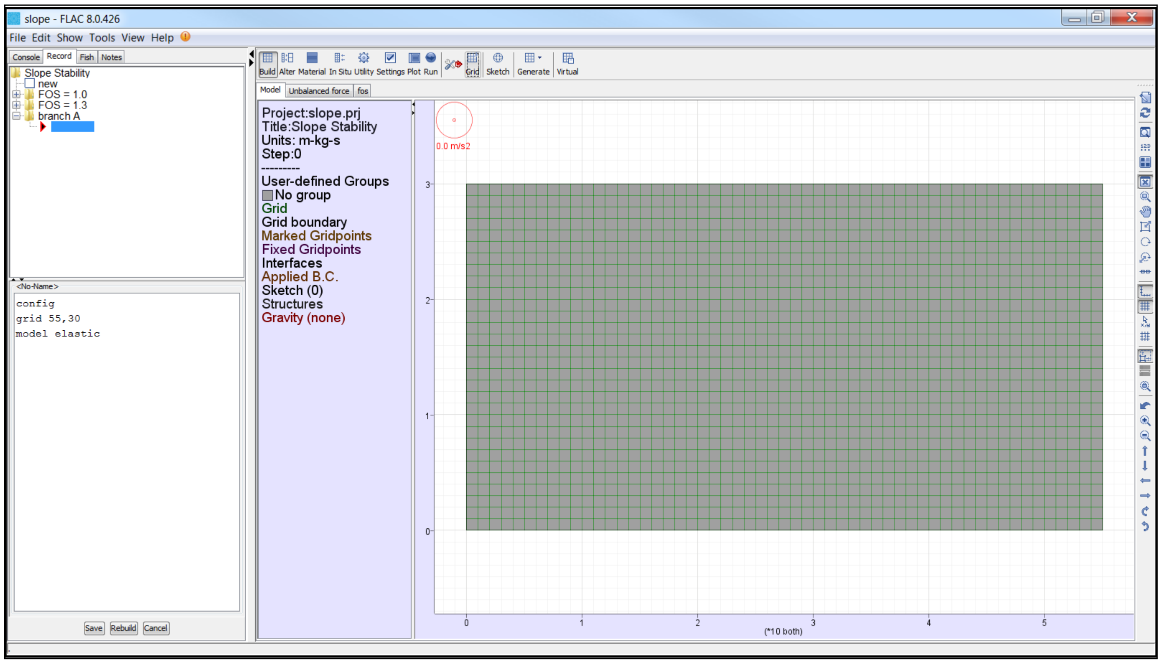
```
=====================================================================
model null
gen 0.0,0.0 0.0,3.0 5.0,3.0 5.0,0.0 i=1,11 j=1,7
gen 5.0,0.0 5.0,3.0 30.0,3.0 30.0,0.0 i=11,56 j=1,7
gen 5.0,3.0 Xtop,13.0 30.0,13.0 30.0,3.0 i=11,56 j=7,31
group 'silty clay' i=1,10 j=1,6
group 'silty clay' i=11,55 j=1,6
group 'silty clay' i=11,55 j=7,30
model mohr group 'silty clay'
prop density=2000.0 bulk=1E8 shear=3E7 group 'silty clay'
prop cohesion=12380.0 friction=20.0        group 'silty clay'
prop dilation=0.0 tension=0.0              group 'silty clay'
fix x i=1 j=1,7
fix x y i=1,11 j=1
fix x i=56 j=1,7
fix x y i=11,56 j=1
fix x i=56 j=7,31
set gravity=10.0
solve fos no_restore file slope_fos_1_3.fsv
=====================================================================
```

Before creating the *FISH* function, a new branch is created in the *FLAC* project tree in the *Record* pane. The branch is created by first double-clicking on the *new* branch name, and then pressing [Follow] to create a new branch. A 55 x 30 zone grid is created using the [Build]/[Grid] tool, as shown in Figure 4.14.

The **slope_angle** *FISH* function (Example 4.2) with the embedded *FLAC* commands (Example 4.3) is now created in the *Fish editor* pane. See Figure 4.15. Check "Enable local record" to activate the *Fish editor*. The function is saved to a file named "slope_angle.fis" by pressing the save button in the toolbar at the top of the *Fish editor* pane.

The *FISH* function can either be executed from the *Fish editor* or run as a **CALL** file in the *FLAC* model. In this example, the function will be executed from the *Fish editor*.

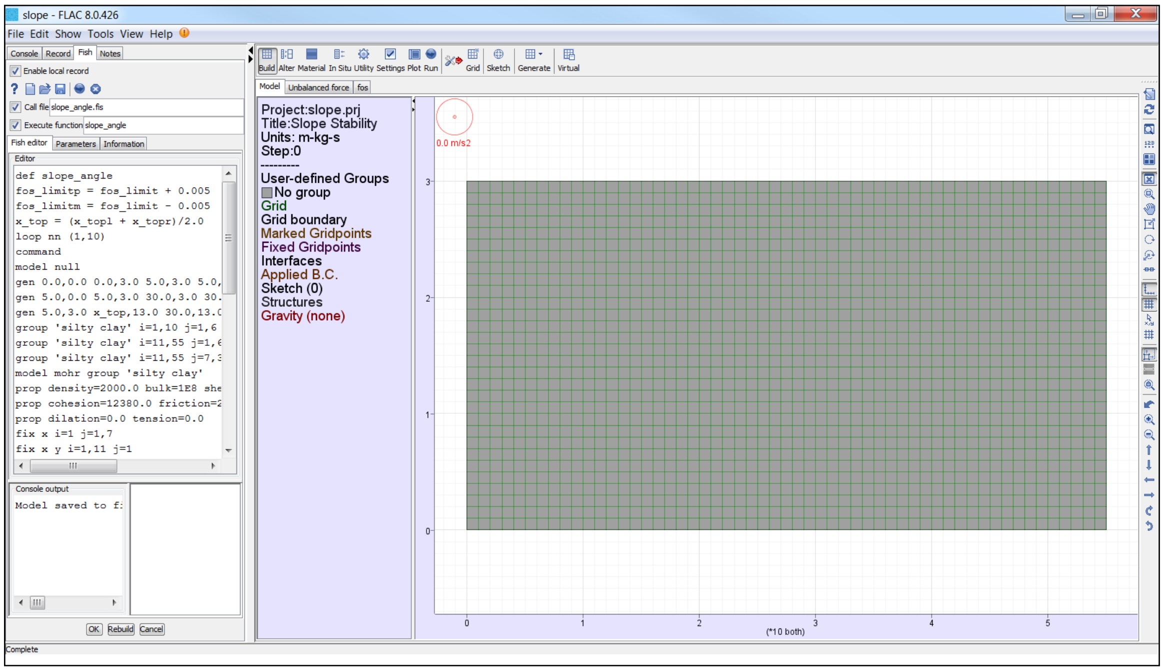
The input values for the *FISH* function (**LXtop**, **RXtop**, and **fos_limit**) are set in the [Parameters] tool of the *Fish editor*, as shown in Figure 4.16. Select [Add] to add the *FISH* name, default value, and description for each parameter.


slope - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Slope Stability
new
FOS = 1.0
FOS = 1.3
branch A
<No-Name>
Save Rebuild Cancel
Build Alter Material In Situ Utility Settings Plot Run Grid Sketch Generate Virtual
Model Unbalanced force fos
0.0 m/s2
(*10 both)

```
config
grid 55,30
model elastic
```

```
Project:slope.prj
Title:Slope Stability
Units: m-kg-s
Step:0
---------
User-defined Groups
No group
Grid
Grid boundary
Marked Gridpoints
Fixed Gridpoints
Interfaces
Applied B.C.
Sketch (0)
Structures
Gravity (none)
```

Figure 4.14 Create a 55 x 30 zone grid to be adjusted by the “slope_angle.fis” *FISH* function.


slope - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Fish editor Parameters Information
Editor
Console output
OK Rebuild Cancel
Build Alter Material In Situ Utility Settings Plot Run Grid Sketch Generate Virtual
Model Unbalanced force fos
0.0 m/s2
(*10 both)
Complete

✓ Enable local record

✓ Call file slope_angle.fis

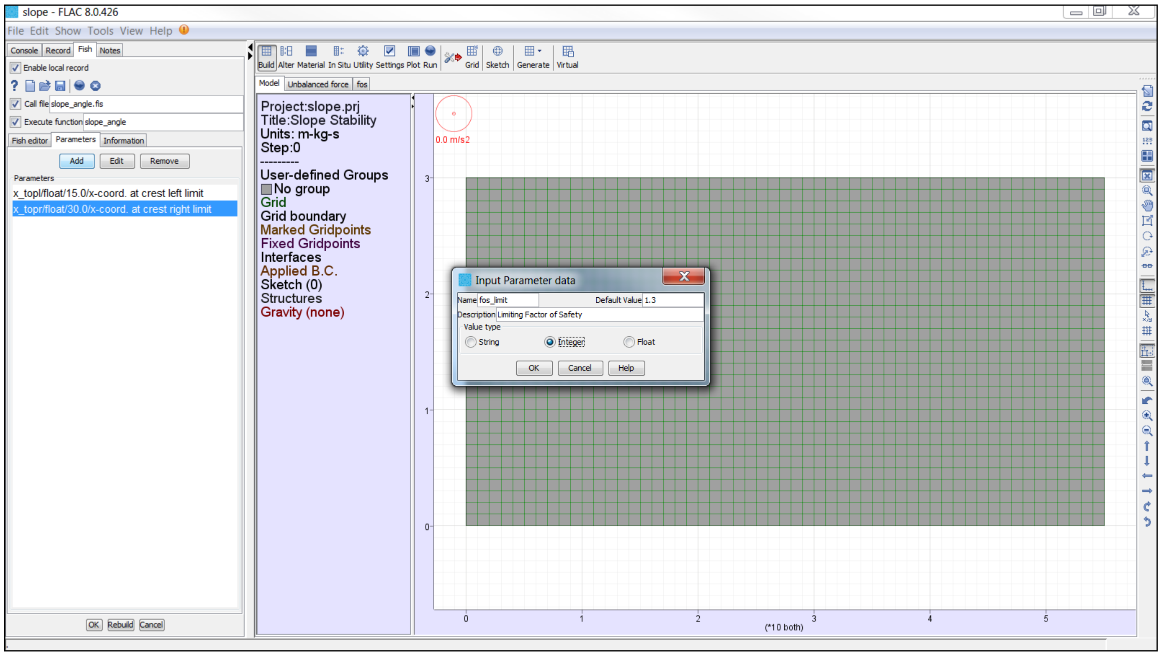
✓ Execute function slope_angle

```
def slope_angle
fos_limitp = fos_limit + 0.005
fos_limitm = fos_limit - 0.005
x_top = (x_topl + x_topr)/2.0
loop nn (1,10)
command
model null
gen 0.0,0.0 0.0,3.0 5.0,3.0 5.0,
gen 5.0,0.0 5.0,3.0 30.0,3.0 30.
gen 5.0,3.0 x_top,13.0 30.0,13.
group 'silty clay' i=1,10 j=1,6
group 'silty clay' i=11,55 j=1,
group 'silty clay' i=11,55 j=7,
model mohr group 'silty clay'
prop density=2000.0 bulk=1E8 she
prop cohesion=12380.0 friction=
prop dilation=0.0 tension=0.0
fix x i=1 j=1,7
fix x y i=1,11 j=1
```

```
Model saved to f
```

```
Project:slope.prj
Title:Slope Stability
Units: m-kg-s
Step:0
---------
User-defined Groups
No group
Grid
Grid boundary
Marked Gridpoints
Fixed Gridpoints
Interfaces
Applied B.C.
Sketch (0)
Structures
Gravity (none)
```

Figure 4.15 Create “Slope_angle.fis” in the *FISH editor* pane.


slope - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Build Alter Material In Situ Utility Settings Plot Run Grid Sketch Generate Virtual
Model Unbalanced force fos
Fish editor Parameters Information
Add Edit Remove
OK Rebuild Cancel
0.0 m/s2
(*10 both)

[✓] Enable local record
[✓] Call file slope_angle.fis
[✓] Execute function slope_angle

Parameters
x_topl/float/15.0/x-coord. at crest left limit
x_topr/float/30.0/x-coord. at crest right limit

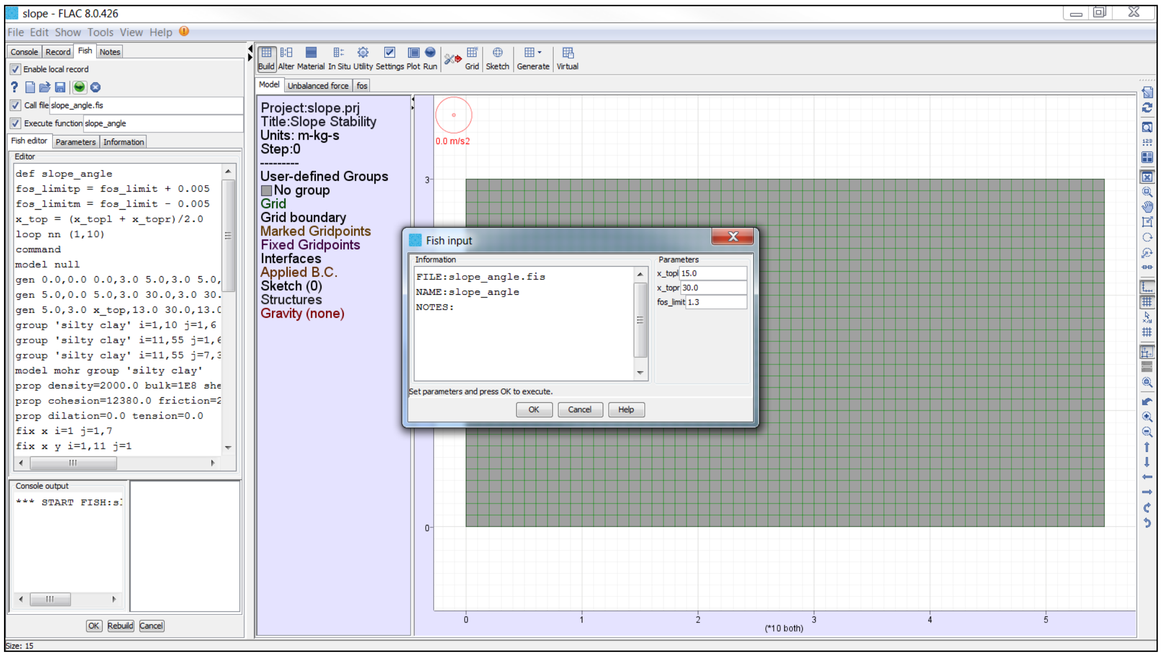
Project:slope.prj
Title:Slope Stability
Units: m-kg-s
Step:0

User-defined Groups
No group
Grid
Grid boundary
Marked Gridpoints
Fixed Gridpoints
Interfaces
Applied B.C.
Sketch (0)
Structures
Gravity (none)

Input Parameter data
Name fos_limit Default Value 1.3
Description Limiting Factor of Safety
Value type: String [✓ Integer] Float
OK Cancel Help

Figure 4.16 Select [Parameters] to add *FISH* input parameters in the *Fish editor*.

Press the button to execute the *FISH* function. A *FISH* input dialog opens and the *FISH* parameters **LXxtop** = 15.0, **RXtop** = 30.0, and **fos_limit** = 1.3 are shown. See Figure 4.17. When [OK] is pressed, the function is executed and the calculation begins.


slope - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Build Alter Material In Situ Utility Settings Plot Run Grid Sketch Generate Virtual
Model Unbalanced force fos
Fish editor Parameters Information
Editor
Console output
OK Rebuild Cancel
0.0 m/s2
(*10 both)
Size: 15

[✓] Enable local record
[✓] Call file slope_angle.fis
[✓] Execute function slope_angle

```
def slope_angle
fos_limitp = fos_limit + 0.005
fos_limitm = fos_limit - 0.005
x_top = (x_topl + x_topr)/2.0
loop nn (1,10)
command
model null
gen 0.0,0.0 0.0,3.0 5.0,3.0 5.0,
gen 5.0,0.0 5.0,3.0 30.0,3.0 30.
gen 5.0,3.0 x_top,13.0 30.0,13.0
group 'silty clay' i=1,10 j=1,6
group 'silty clay' i=11,55 j=1,6
group 'silty clay' i=11,55 j=7,3
model mohr group 'silty clay'
prop density=2000.0 bulk=1E8 she
prop cohesion=12380.0 friction=2
prop dilation=0.0 tension=0.0
fix x i=1 j=1,7
fix x y i=1,11 j=1
```

```
*** START FISH:sl
```

Project:slope.prj
Title:Slope Stability
Units: m-kg-s
Step:0

User-defined Groups
No group
Grid
Grid boundary
Marked Gridpoints
Fixed Gridpoints
Interfaces
Applied B.C.
Sketch (0)
Structures
Gravity (none)

Fish input
Information
```
FILE:slope_angle.fis
NAME:slope_angle
NOTES:
```
Parameters
x_topl 15.0
x_topr 30.0
fos_limit 1.3

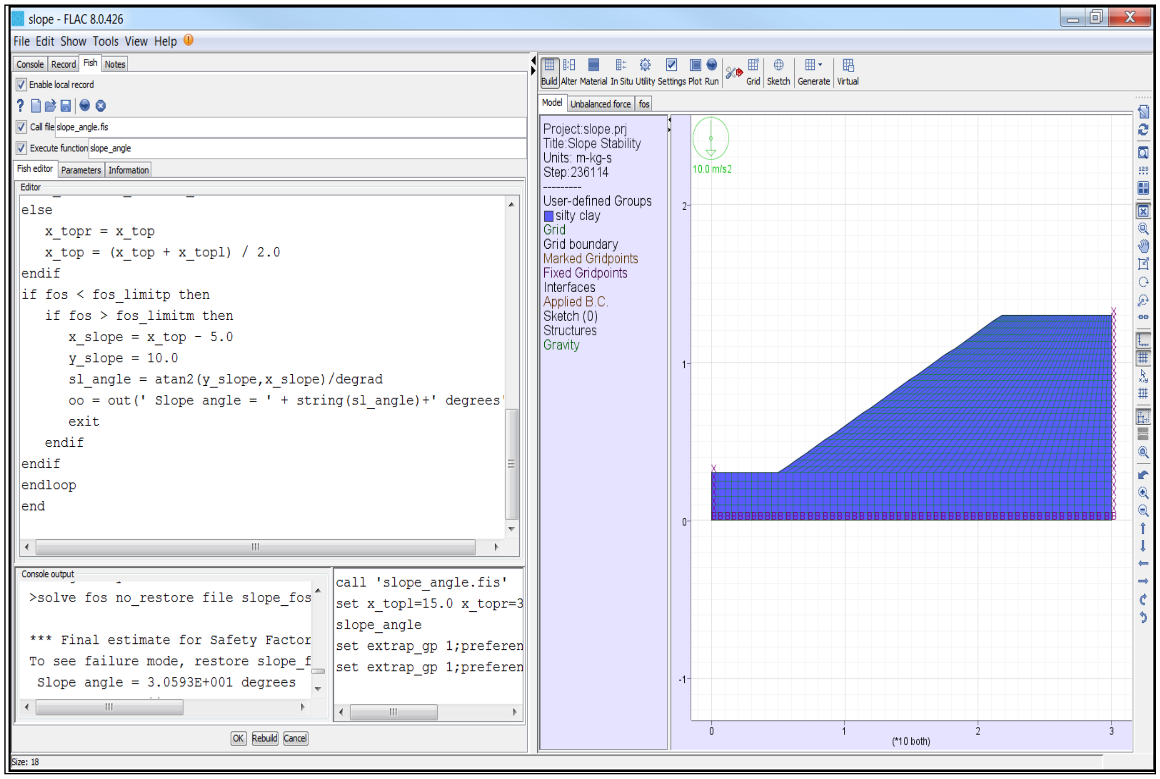
Set parameters and press OK to execute.
OK Cancel Help

Figure 4.17 Execute the "slope_angle.fis" *FISH* fuction and change input for LXtop, RXtop, and fos_limit in the *FISH* input dialog.

4.4 Results

The **slope_angle** function performs six **SOLVE fos** calculations (type **PRINT nn** in the *Console* pane to print the number of iterations). When the run stops, the slope at failure for FoS = 1.3 is plotted, and the slope angle is printed in the *Console output*, as shown in Figure 4.18. The slope angle that produces a factor of safety of 1.3 is 30.6°.

Press [OK] and return to the *Record* pane, and then save the state as "slope_fos_1_3.fsv" in the project tree. This state is the failure state at FoS = 1.3. The failure plane can be viewed selecting the [Plot]/[Model] tool and picking the [Factor of Safety] plot item; see Figure 4.19.



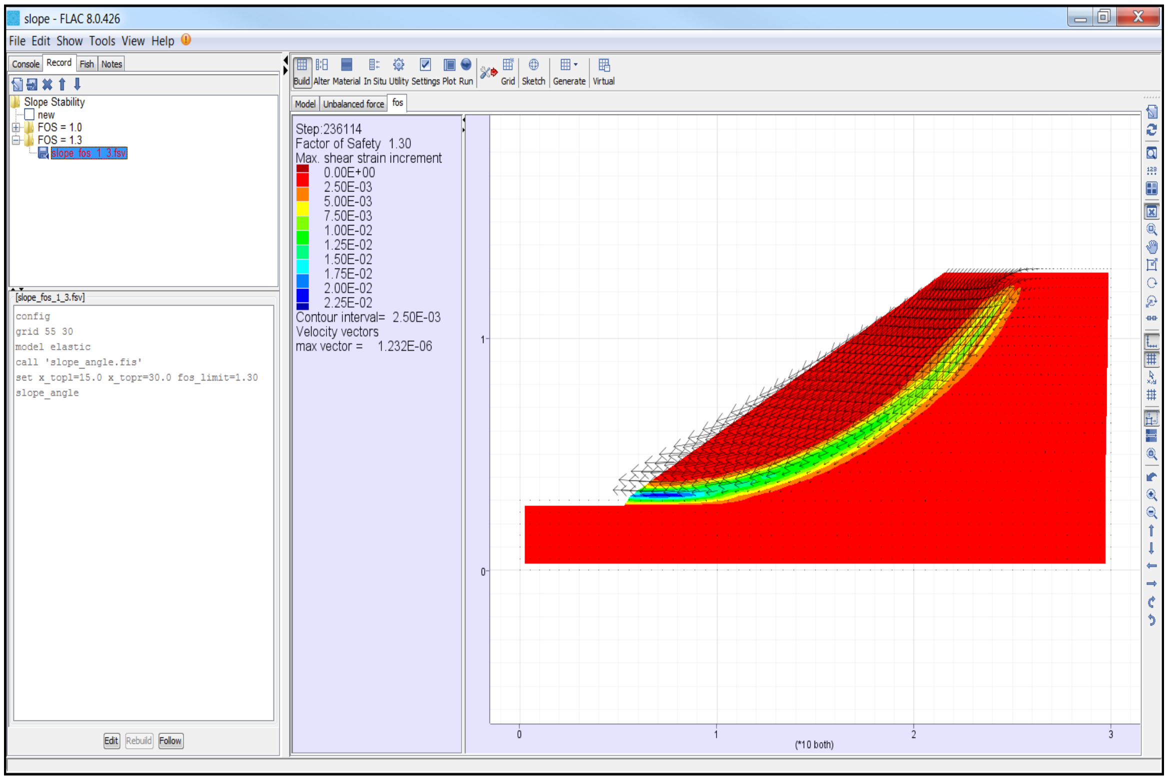
```
else
   x_topr = x_top
   x_top = (x_top + x_topl) / 2.0
endif
if fos < fos_limitp then
   if fos > fos_limitm then
      x_slope = x_top - 5.0
      y_slope = 10.0
      sl_angle = atan2(y_slope,x_slope)/degrad
      oo = out(' Slope angle = ' + string(sl_angle)+' degrees'
      exit
   endif
endif
endloop
end
```

```
 >solve fos no_restore file slope_fos

 *** Final estimate for Safety Factor
 To see failure mode, restore slope_f
  Slope angle = 3.0593E+001 degrees
```

```
call 'slope_angle.fis'
set x_topl=15.0 x_topr=3
slope_angle
set extrap_gp 1;preferen
set extrap_gp 1;preferen
```

Figure 4.18 The slope angle corresponding to FoS = 1.3 is printed in the *Console output*.



slope - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Slope Stability
new
FOS = 1.0
FOS = 1.3
slope_fos_1_3.fsv
Build Alter Material In Situ Utility Settings Plot Run
Grid Sketch Generate Virtual
Model Unbalanced force fos
(*10 both)

[slope_fos_1_3.fsv]

```
config
grid 55 30
model elastic
call 'slope_angle.fis'
set x_topl=15.0 x_topr=30.0 fos_limit=1.30
slope_angle
```

Step:236114
Factor of Safety 1.30
Max. shear strain increment

🟥	0.00E+00
🟥	2.50E-03
🟧	5.00E-03
🟨	7.50E-03
🟩	1.00E-02
🟩	1.25E-02
🟩	1.50E-02
🟦	1.75E-02
🟦	2.00E-02
🟦	2.25E-02

Contour interval= 2.50E-03
Velocity vectors
max vector = 1.232E-06

Figure 4.19 Factor of safety result (FoS = 1.3).

4.5 On Your Own

1. The agreement between *FLAC* results and the solution from Chen (2007) can be improved by increasing the number of zones. Re-run this problem with the average total zones (horizontal) set to [Extrafine] zones in the *Zoning options* dialog. Try to reproduce the result shown in Figure 4.13 and note the difference in computation time for the [Extra fine] zoned model compared to the [Medium] zoned model.

2. The parameter study can be cloned to determine the slope angle associated with different factors of safety. Repeat Stage 2 to determine the slope angle that results in a FoS = 1.2.

Chapter 5
Adding Complexity: A Shallow Tunnel in Soft Ground

5.1 Background

Various methods of solution have been offered for the evaluation of the effects at the surface of the construction of a shallow tunnel in soft ground. The solution of this problem is of particular importance in urban areas where minimizing the impact on existing structures and services often becomes a major constraint in design. Rankin (1988)[1] summarizes criteria for the assessment of risk to a structure due to tunneling operations and suggests strategies for the various risk categories. These concepts, based on the surface settlement trough shape being approximated as an inverted Gaussian distribution curve (as proposed by Peck [1969][2] and derived largely from observations of tunnels in overconsolidated clays), have been used quite successfully in generalized two- and three-dimensional forms.

The surface settlement trough shape is given as a function of the depth of the tunnel axis below the surface, a trough width parameter ***K***, dependent on the soil type, and a surface volume loss ***V***, often expressed as a percentage of the tunnel area measured at a specified location along the tunnel and related to the construction method and soil type. Figure 5.1 (Yeats, 1985[3]) shows the general concept. Assuming that the tunnel geometry is constrained, the only variable under design control is the surface volume loss ***V*** above a location along the tunnel axis. By altering construction techniques, the designer can influence the volume loss ***V*** and, consequently, the effect of the tunnel at the surface.

Generally the soil stratigraphy and behavior are not taken into account in this type of analysis, and material properties are subsumed in ***K*** and ***V***.

At any point, *y*, transverse from the tunnel centerline, the vertical settlement, *w*, in the fully developed trough is given by:

$$w = \mathrm{W}_{\max} \exp(-y^2/2i^2)$$

where

$$\mathrm{W}_{\max} = 0.0125\ V r^2/i$$
$$i = K z_o$$

and the symbols are as shown in Figure 5.1. *r* is the tunnel radius at the location along the tunnel axis where the surface settlement is to be determined.

This is very much a three-dimensional problem. For simple ground conditions, by using the axisymmetric capabilities in *FLAC*, the behavior in the running direction can be analyzed and data about the relation between closure and distance between lining and face can be obtained. However, the proximity of the ground surface and the particular stratigraphy in this example make the system non-axisymmetric, leaving us unable to determine this relation with two-dimensional tools.

1. Rankin W. J. "Ground Movements Resulting from Urban Tunnelling: Predictions and Effects," in ***Engineering Geology of Underground Movements (Proceedings of the 23rd Annual Conference of the Engineering Group of the Geological Sociaty, Nottingham, England, 1998)***, pp. 79–92 (1988).
2. Peck R. B. "Deep Excavations and Tunneling in Soft Ground," in ***Proceedings of the 7th International Conference on Soil Mechanics and Foundation Engineering (Mexico City, 1969)***, Vol. 3, pp. 225–290 (1969).
3. Yeats J. "The Response of Buried Pipelines to Ground Movements Caused By Tunneling in Soil," in ***Ground Movements and Structures***, pp. 145–160. J. D. Geddes, Ed. Plymouth: Pentech Press (1985).

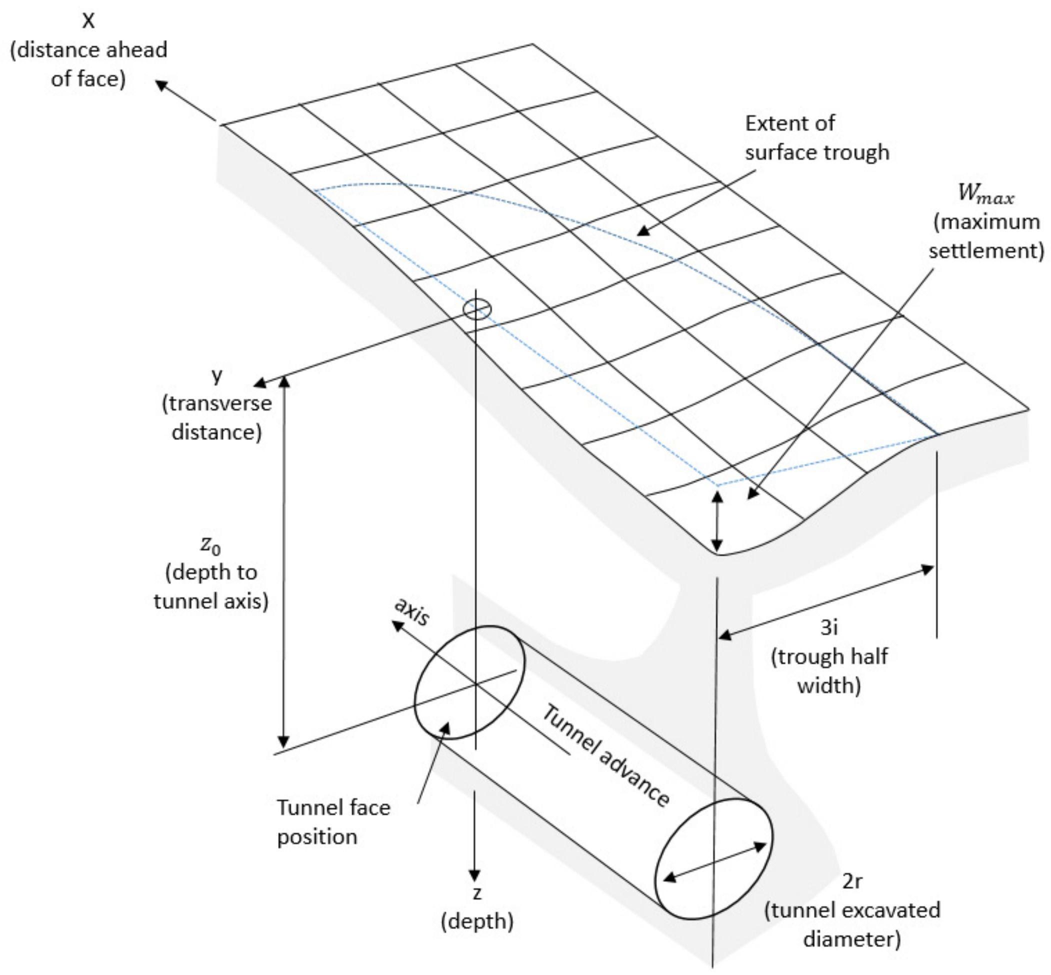


Figure 5.1 Geometry of tunnel and surface trough (Yeats, 1985).

Even though this is a three-dimensional problem, we can examine, in two dimensions, important aspects of the tunnel construction process, i.e., the amount of tunnel closure that occurs prior to installation of ground support and the effects of the closure and relaxation of stresses around the tunnel on the lining loads and on surface settlement. The distance behind the tunnel face to the location where support is installed affects the tunnel closure and the tunnel loads. This is typically expressed as a longitudinal tunnel-displacement profile (LDP) that relates tunnel displacement to distance from the excavation face. The LDP is commonly used in preliminary 2D analysis of tunnel closure and support design to estimate the influence of the location of the tunnel face on the tunnel loads.

The LDP cannot be calculated from a two-dimensional, plane-strain analysis. The profile can be calculated using 2D axisymmetric models if the stress conditions are uniform and the tunnel cross-section is circular. However, in most cases, a full three-dimensional model is required to calculate the LDP for complex geometries and loading conditions.

Figure 5.2 illustrates the characteristics of the LDP based upon advancement of a 10-m diameter tunnel at depth in a given rock type. The figure illustrates that a portion of the tunnel closure occurs before the tunnel advances past a specific point, and the tunnel continues to displace inward as the tunnel advances beyond that point.

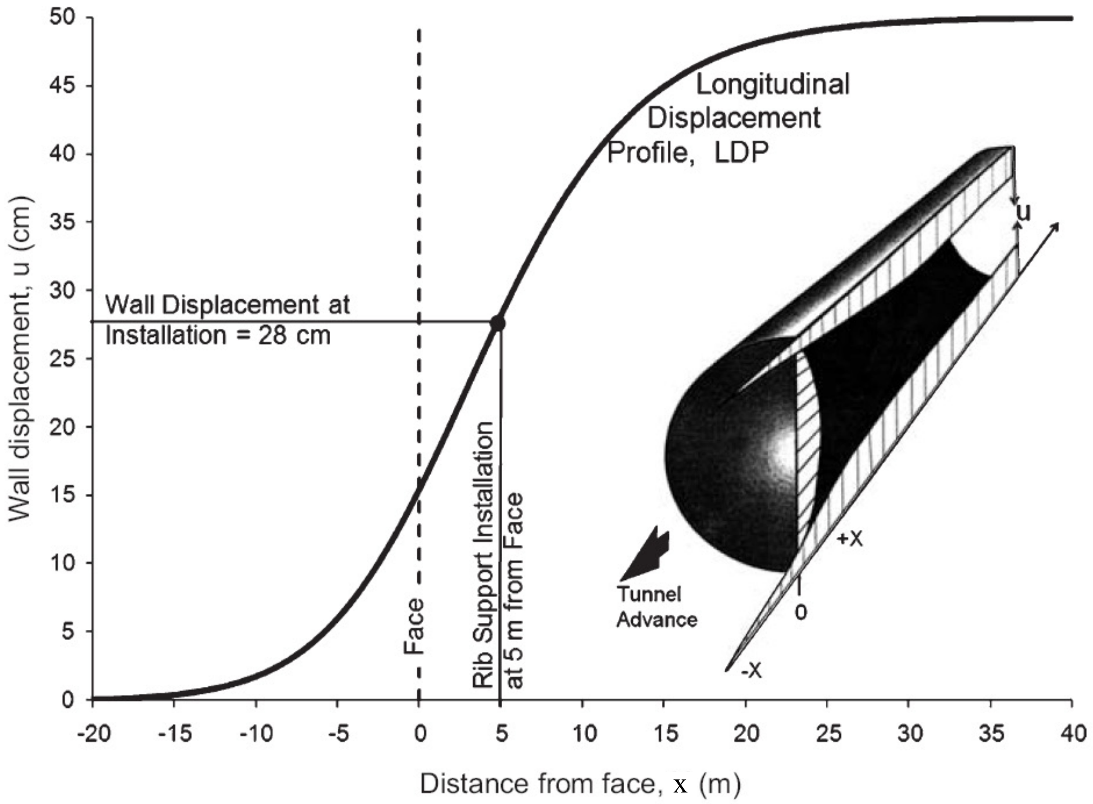


Figure 5.2 Longitudinal displacement profile (Vlachopoulos and Diederichs, 2009[1]).

The LDP is used in 2D plane-strain models to estimate the three-dimensional effect of tunnel advancement and to determine the appropriate timing for installation of support. Two methods are commonly employed in two-dimensional analysis to simulate the effect of tunnel advancement on lining loads. The first method, called the **core-replacement method,** uses a soft, elastic inclusion placed within the tunnel region to limit the closure, and, by successively reducing the modulus of the inclusion, a characteristic curve relating closure to modulus can be developed. The core-replacement characteristic curve is then used with the LDP to define an inclusion modulus that corresponds to a tunnel advancement, at which point the tunnel support is installed. An application of this approach is presented by Hoek et al. (2008).[2]

A second method, called the **convergence-confinement method**, uses a characteristic curve, a **ground reaction curve**, that is developed by progressively reducing the internal pressure within the excavation region and plotting this support pressure versus the tunnel closure. The incremental reduction of the internal pressure represents the effect of the advancing tunnel face. By knowing the installed support distance from the tunnel face and using the LDP, the amount of tunnel deformation that occurs prior to support installation can be calculated by relaxing the pressure to a given level corresponding to that deformation.

1. Vlachopoulos, N. and M.S. Diederichs. "Improved Longitudinal Displacement Profiles for Convergence Confinement Analysis of DeepTunnels," *Rock Mech. Rock Engr.* ***42***(2), 131–146 (2009).

2. Hoek, E., C. Carranza-Torres, M. Diederichs and B. Corkum. "The 2008 Kersten Lecture: Integration of Geotechnical and Structural Design in Tunneling," presented at the University of Minnesota 56th Annual Geotechnical Engineering Conference, Minneapolis, Minnesota, February (2008).

Vlachopoulos and Diederichs (2009) present a ground reaction curve plotted with an LDP in Figure 5.3. The internal pressure indicates the amount of support resistance required to prevent additional tunnel closure. Figure 5.3 also shows a support reaction curve for a tunnel liner installed at the tunnel face and illustrates the effect of the support. Tunnel sections shown on the figure illustrate the development of the plastic zone with and without support. For additional discussion and guidelines for the convergence-confinement method, see Lorig and Varona (2013)[1].

FLAC in its two-dimensional plane-strain form can be applied to this problem using either the core-replacement or convergence-confinement analysis method, providing good data both at the surface and at the tunnel simultaneously. The second method in particular can easily obtain a ground reaction curve from a *FLAC* analysis because deformations are determined directly from the change in loading in the explicit solution procedure. This is demonstrated in the following example problem.

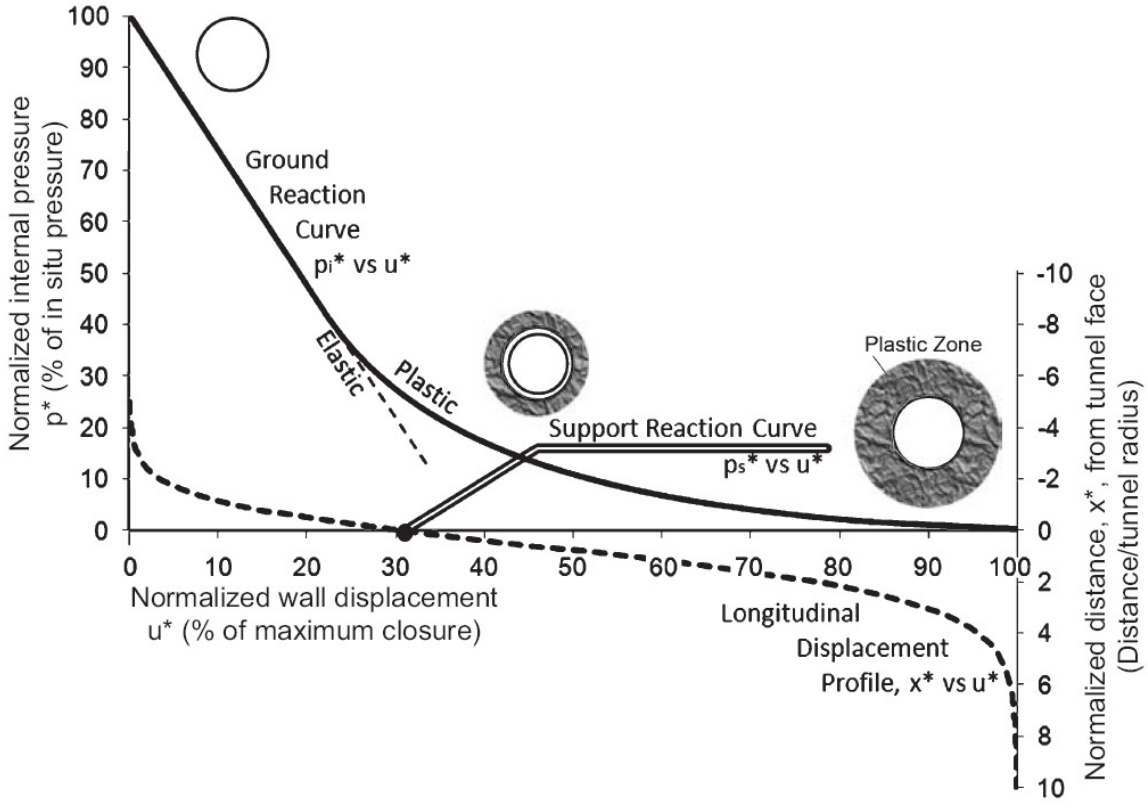


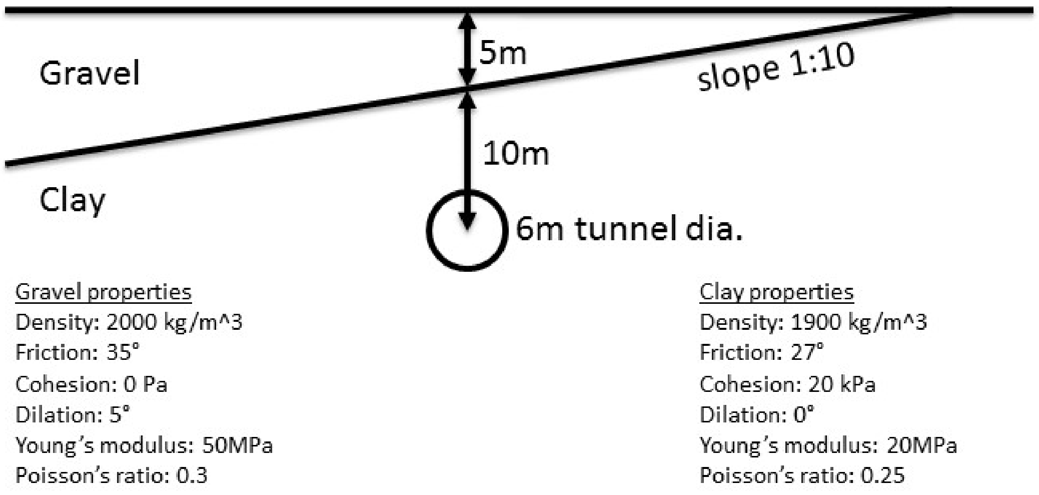
Figure 5.3 Ground reaction curve shown with a support reaction curve for a liner installed at the tunnel face and the relation to an LDP (Vlachopoulos and Diederichs, 2009).

1. Lorig, L. J., and P. Varona. "Guidelines for Numerical Modelling of Rock Support for Mines," in ***Ground Support 2013: Proceedings, 7th International Symposium on Ground Support in Mining and Underground Construction (Perth, May 2013)***, pp. 81–105, Y. Potvin and B. Brady, Eds. Perth: Australian Centre for Geomechanics (2013).

5.2 Problem Description

Difficult geometric conditions have been chosen to illustrate some of the modeling techniques in *FLAC*. This is a three-dimensional problem and is more complex than the previous practical application. There is more than one material involved and the stratigraphy itself adds complications that need to be addressed. A controlled excavation of a tunnel must be performed. An important modeling concern is that the answers must not be influenced by the applied boundary conditions.

The problem geometry and material properties are shown in Figure 5.4. Material properties typically come from laboratory testing, but should be extrapolated to field scale values. In this example, properties are assumed to be in-situ properties. For example, density is the in-situ moist density. For suggestions on selecting properties for a model, see Section 3.7 in the *FLAC* **User's Guide**.



Gravel properties
Density: 2000 kg/m^3
Friction: 35°
Cohesion: 0 Pa
Dilation: 5°
Young's modulus: 50MPa
Poisson's ratio: 0.3

Clay properties
Density: 1900 kg/m^3
Friction: 27°
Cohesion: 20 kPa
Dilation: 0°
Young's modulus: 20MPa
Poisson's ratio: 0.25

Figure 5.4 Problem geometry and material properties.

The following points will be addressed in the solution for this problem.

- The development of the ground reaction curve for the system.
- The examination of the loads in the liner.
- The calculation of a surface profile of vertical displacements.

5.3 Modeling Procedure

A recommended procedure is presented to perform an analysis with *FLAC* to simulate the construction of a shallow tunnel in soft ground. The tunnel is supported by a shotcrete lining that is installed at a distance of 2 m behind the tunnel face.

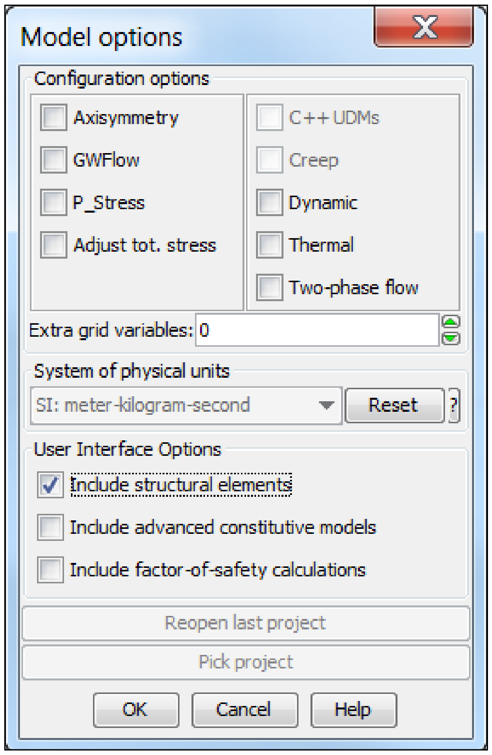
During the process of illustrating this procedure and satisfying the problem requirements, several capabilities of *FLAC* will be demonstrated, including:

- techniques used in setting up model grids;
- the modeling of structured geology;
- the process of excavating a feature;
- boundary condition options;
- tracking the behavior history of selected points;
- the use of *FISH* functions;
- judging the state of equilibrium in a model; and
- the usage of tables to collect and manipulate data.

5.3.1 Start-Up Conditions

When starting this *FLAC* project, the "Include structural elements" option is selected in the *Model options* dialog. See Figure 5.5. The various features of the structural element logic in *FLAC* are now accessible from the graphical interface. A *Structure* tab is added to the modeling stage tabs to access the structural elements tools.

A *FLAC* project, named "tunnel.prj", is created to perform this analysis. Both the ground reaction curve calculation and the tunnel support calculation will be contained within this project.


Model options

Configuration options

- ☐ Axisymmetry
- ☐ GWFlow
- ☐ P_Stress
- ☐ Adjust tot. stress
- ☐ C++ UDMs
- ☐ Creep
- ☐ Dynamic
- ☐ Thermal
- ☐ Two-phase flow

Extra grid variables: 0

System of physical units

SI: meter-kilogram-second

User Interface Options

- ✓ Include structural elements
- ☐ Include advanced constitutive models
- ☐ Include factor-of-safety calculations

Figure 5.5 Select "Include structural elements" to access the structural element logic in *FLAC*.

5.3.2 Position of Model Boundaries

When attempting to model a three-dimensional structure, such as a tunnel, in two dimensions, some simplifications and assumptions need to be made. Specifically, that the model is infinite in the third dimension and that the model boundaries in the plane of analysis do not influence model results. Model geometry and boundary conditions will be defined by the soil and tunnel geometry and the type of analysis to be done. In the case under discussion, the settlement trough at the surface is expected to be about three times as wide as the depth of the tunnel axis on each side of the axis. The boundary should be placed far enough away from the trough edge so as to reduce effects, such as tensions building in the material adjacent to the boundary; perhaps a total of four times the depth will suffice in this case. (This is discussed further in Section 5.3.4.) The model size initially selected is 120 m wide by 45 m deep.

Note that if the interest were only at the tunnel itself, the boundaries could be placed nearer the tunnel. These effects can be tested and the model size optimized. Planes of symmetry can reduce the overall model size with direct benefits in computation speed. In this case, however, the inclined geologic contact precludes this.

5.3.3 Designing the Model Grid

There are a number of conflicting criteria that govern the choice of grid for a model. Some factors will influence model accuracy.

- Finer meshes lead to more accurate results in that they provide a better representation of high-stress gradients.
- Accuracy increases as zone aspect ratios tend to unity.
- If different zone sizes are needed, the more gradual the change from the smallest to the largest, the better the results.

However, as the mesh is made finer and the number of zones increases, more computer memory is required and the computation time lengthens. Improving the mesh geometry by adjusting the zone size will certainly improve memory efficiency (which is not as important today) and will reduce computation time (which is important). It will also take some time to plan and set up. The more complicated the geometry gets, the greater the scope for improvement in memory use and calculation time in moving from a "brute force" approach (i.e., single sized elements in a large uniform mesh) to a more rigorous solution.

In the discussion on boundaries, it was concluded that the model should be 120 m wide by 45 m deep. It is apparent that the greater the number of linear segments used to model a circle, the smoother the circle becomes. Therefore, a fairly dense mesh is required to provide the tunnel outline. However, extending such a mesh to the boundaries will result in a large number of zones and, consequently, long computation times. Figure 5.6 shows the results of increasing mesh density in a 12 m by 12 m block that contains a 6 m circle at the center.

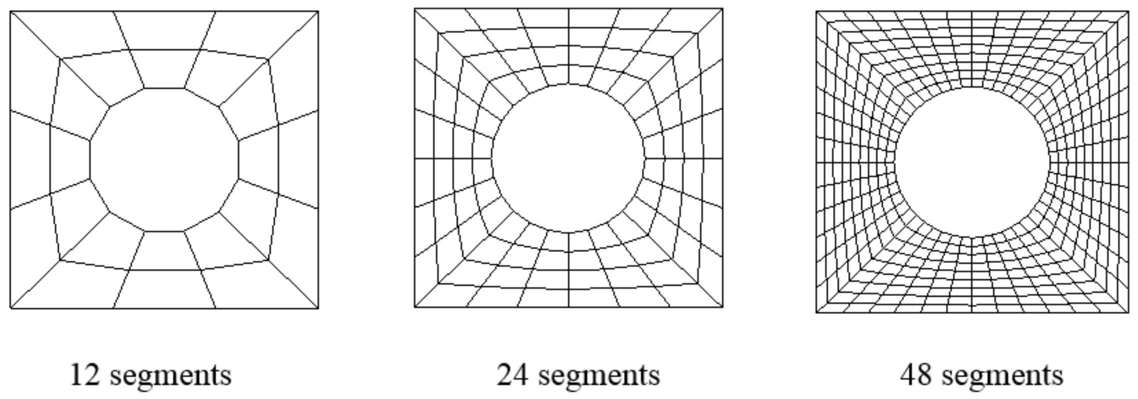


Figure 5.6 Influence of mesh size on geometric representation.

A mesh with the tunnel periphery defined by 48 segments should be created in order to provide an accurate representation of the stress gradient around the tunnel. Note that all zones are four sided in Figure 5.6. This provides better control to produce a zone aspect ratio close to unity in the vicinity of the tunnel. (It is possible to create a grid with three-sided zones in *FLAC*. However, it is recommended that four-sided zones be used for a more accurate solution.) The optimal grid should then gradually change from the fine mesh around the tunnel to a more coarse mesh farther away from the tunnel.

The design of an optimal grid that addresses the above factors for a given *FLAC* analysis can be quite time consuming because commands must be executed to generate the grid each time a different grid configuration is created. The "virtual grid" mode provides an easy way to design the model grid before creating commands to send to *FLAC*. The virtual grid is accessed from the [Build] tool tab. When [Build] is pressed, four tool tabs become active, as shown in Figure 5.7. These tools are discussed in detail in Section 1.2.1 of the *FLAC* **Graphical User Interface Reference**.

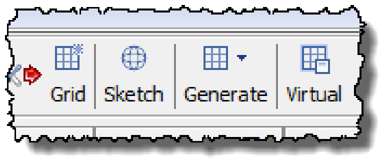


Figure 5.7 [BUILD] tool tabs.

The [Generate] tool is used to create geometries of arbitrary shape. When [Generate] is pressed, a geometries menu opens, as shown in Figure 5.8. Several tools are available to create different shapes. The [Geometry builder] tool is used in this example because it can be applied easily to create both the boundary between the gravel and clay and the circular tunnel boundary.

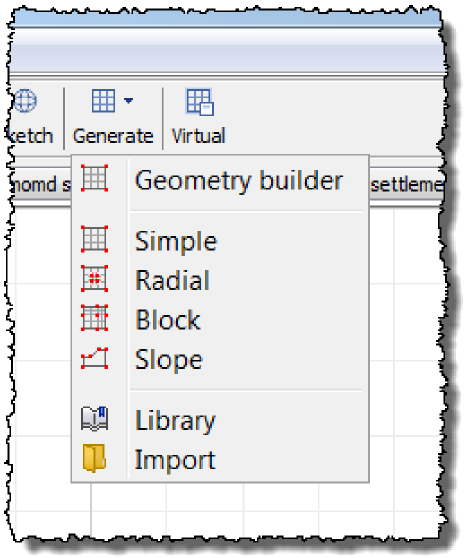


Figure 5.8 [GENERATE] tool geometries menu.

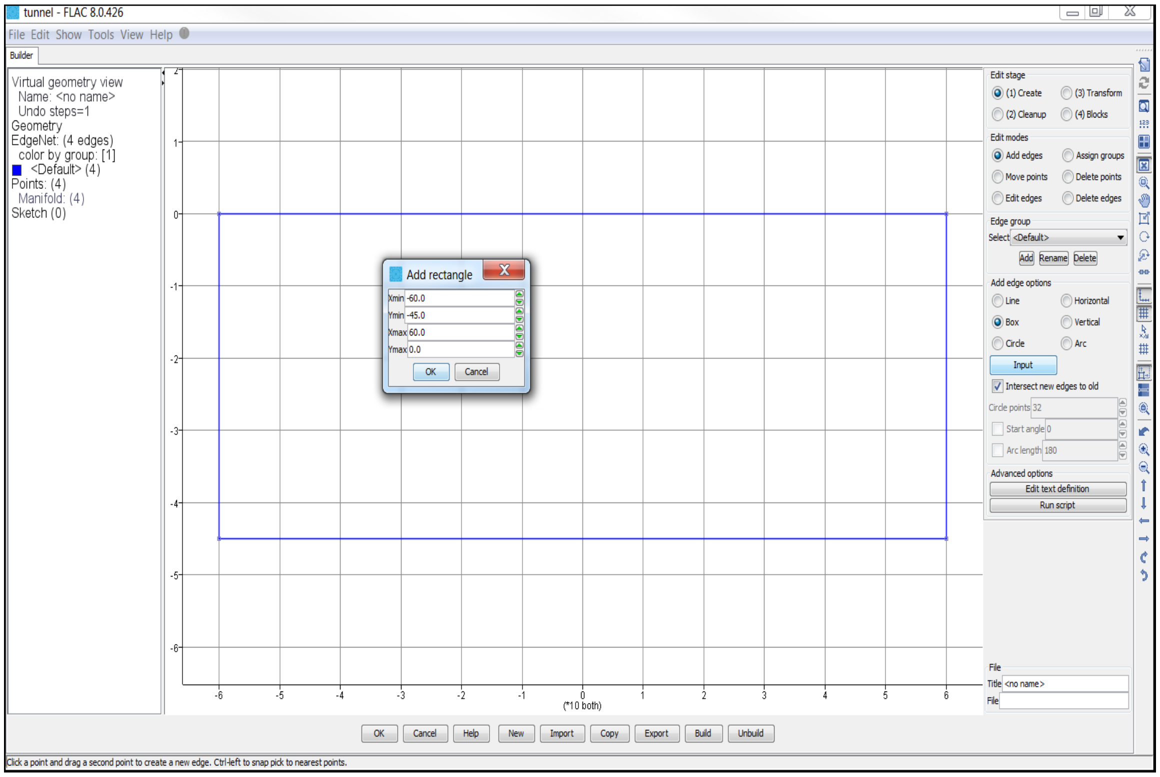
When the [Geometry builder] tool is entered, several drawing edit modes are available. In Figures 5.9 through 5.11, the [Add edges] mode is used to draw first the outer boundary for the model (Figure 5.9), then the periphery of the 6 m diameter tunnel (Figure 5.10) with centroid at elevation -15.0, and finally the boundary between the gravel and clay (Figure 5.11).

The [Geometry builder] tool can be used to either sketch the problem geometry manually or import a user-defined drawing, such as a DXF file.

Construction lines are now added to the model to provide control over zoning. First, a box is drawn with construction lines around the circular tunnel. This will facilitate the creation of a radial mesh of quadrilateral zones around the tunnel, as shown in Figure 5.6. Additional construction lines are added until the model region is composed entirely of four-sided blocks, as shown in Figure 5.12.

The location of the construction lines is arbitrary; the goal is to provide a means to increase zone size (and thus reduce the number of zones) as we move away from the tunnel region. As Figure 5.12 shows, three layers of blocks are created around the tunnel. The zoning can thus be manually changed for each block.

The model region must be composed entirely of four-sided blocks before any mesh adjustment can be performed. The [Cleanup] and [Blocks] edit stages are used, as described in Section 1.2.1.6 of the *FLAC* **Graphical User Interface Reference**, to facilitate the creation of model blocks. [Build] is pressed to create the block sub-division. For the construction line division shown in Figure 5.12, the model is sub-divided into 35 "quad blocks."



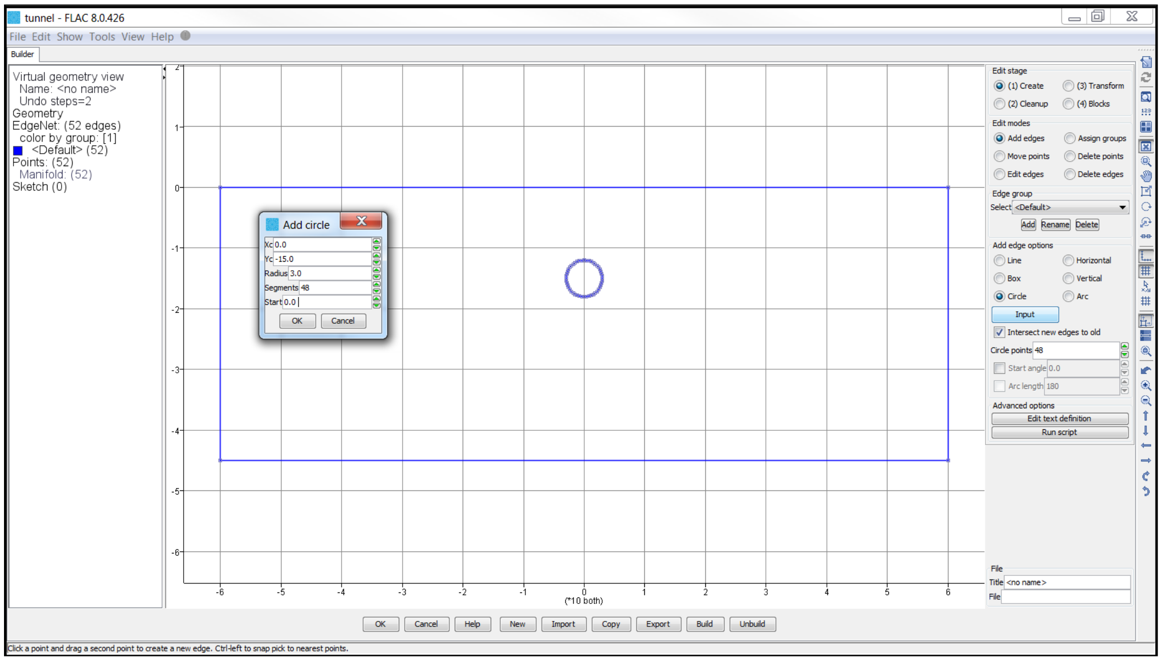
Virtual geometry view
Name: <no name>
Undo steps=1
Geometry
EdgeNet: (4 edges)
color by group: [1]
<Default> (4)
Points: (4)
Manifold: (4)
Sketch (0)

Edit stage: [✓ (1) Create]
Edit modes: [✓ Add edges]
Edge group Select: <Default>
Add edge options: [✓ Box]
[✓ Intersect new edges to old]
Circle points 32
Start angle 0
Arc length 180
Title <no name>

Add rectangle
Xmin -60.0
Ymin -45.0
Xmax 60.0
Ymax 0.0

Click a point and drag a second point to create a new edge. Ctrl-left to snap pick to nearest points.

Figure 5.9 Draw the outer model boundary with the [Box] edit mode. (Xmin = -60.0, Ymin = -45.0, Xmax = 60.0, Ymax = 0.0).



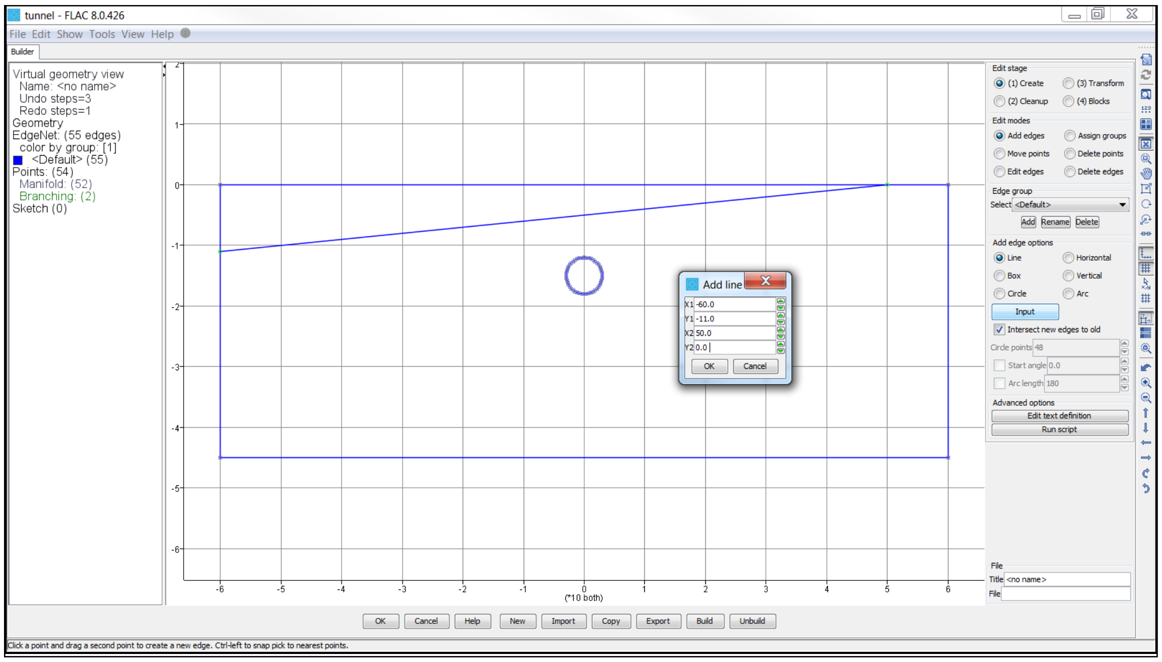
Virtual geometry view
Name: <no name>
Undo steps=2
Geometry
EdgeNet: (52 edges)
color by group: [1]
■ <Default> (52)
Points: (52)
Manifold: (52)
Sketch (0)

Add circle
Xc 0.0
Yc -15.0
Radius 3.0
Segments 48
Start 0.0

Edit stage: [✓ (1) Create] (2) Cleanup (3) Transform (4) Blocks
Edit modes: [✓ Add edges] Move points Edit edges Assign groups Delete points Delete edges
Edge group: Select <Default>
Add edge options: Line Box [✓ Circle] Horizontal Vertical Arc
[✓ Intersect new edges to old]
Circle points 48
Start angle 0.0
Arc length 180
Title <no name>

Click a point and drag a second point to create a new edge. Ctrl-left to snap pick to nearest points.

Figure 5.10 Draw the tunnel periphery with the [Circle] edit mode. (Xc = 0.0, Yc = -15.0, Radius = 3.0, Segments = 48).



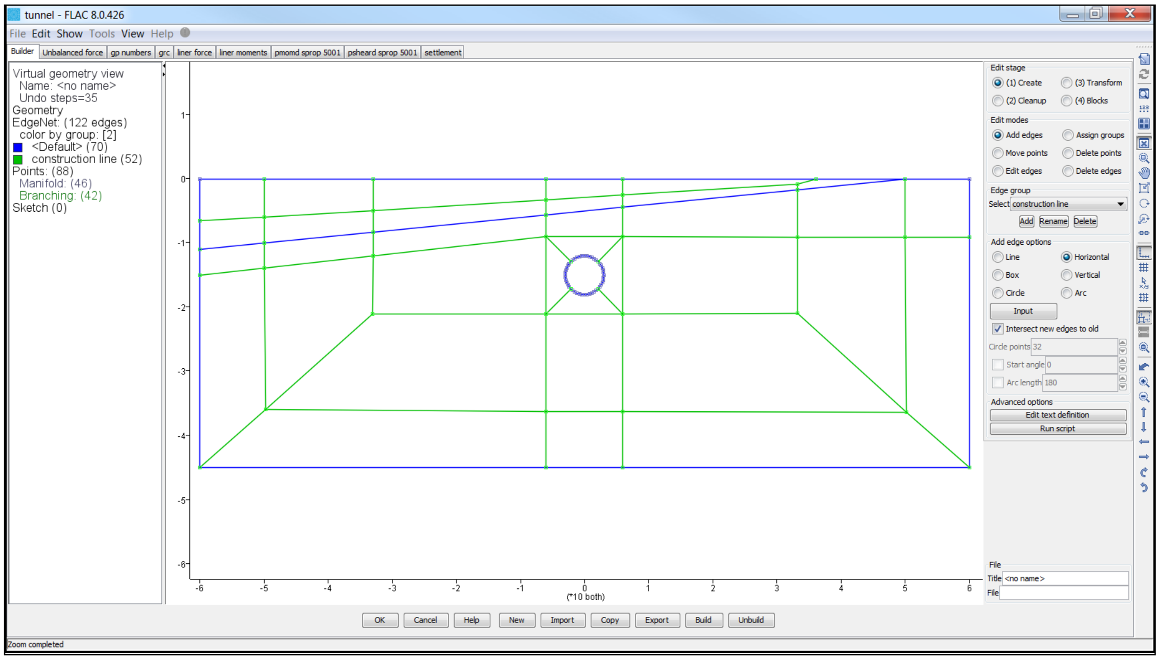
Virtual geometry view
Name: <no name>
Undo steps=3
Redo steps=1
Geometry
EdgeNet: (55 edges)
color by group: [1]
■ <Default> (55)
Points: (54)
Manifold: (52)
Branching: (2)
Sketch (0)

Add line
X1 -60.0
Y1 -11.0
X2 50.0
Y2 0.0

Edit stage: [✓ (1) Create] (2) Cleanup (3) Transform (4) Blocks
Edit modes: [✓ Add edges] Move points Edit edges Assign groups Delete points Delete edges
Edge group: Select <Default>
Add edge options: [✓ Line] Box Circle Horizontal Vertical Arc
[✓ Intersect new edges to old]
Circle points 48
Start angle 0.0
Arc length 180
Title <no name>

Click a point and drag a second point to create a new edge. Ctrl-left to snap pick to nearest points.

Figure 5.11 Draw the gravel-clay boundary with the [Line] edit mode. (X1 = -60.0, Y1 = -11.0, X2 = 50.0, Y2 = 0.0).



Virtual geometry view
Name: <no name>
Undo steps=35
Geometry
EdgeNet: (122 edges)
color by group: [2]
🟦 <Default> (70)
🟩 construction line (52)
Points: (88)
Manifold: (46)
Branching: (42)
Sketch (0)

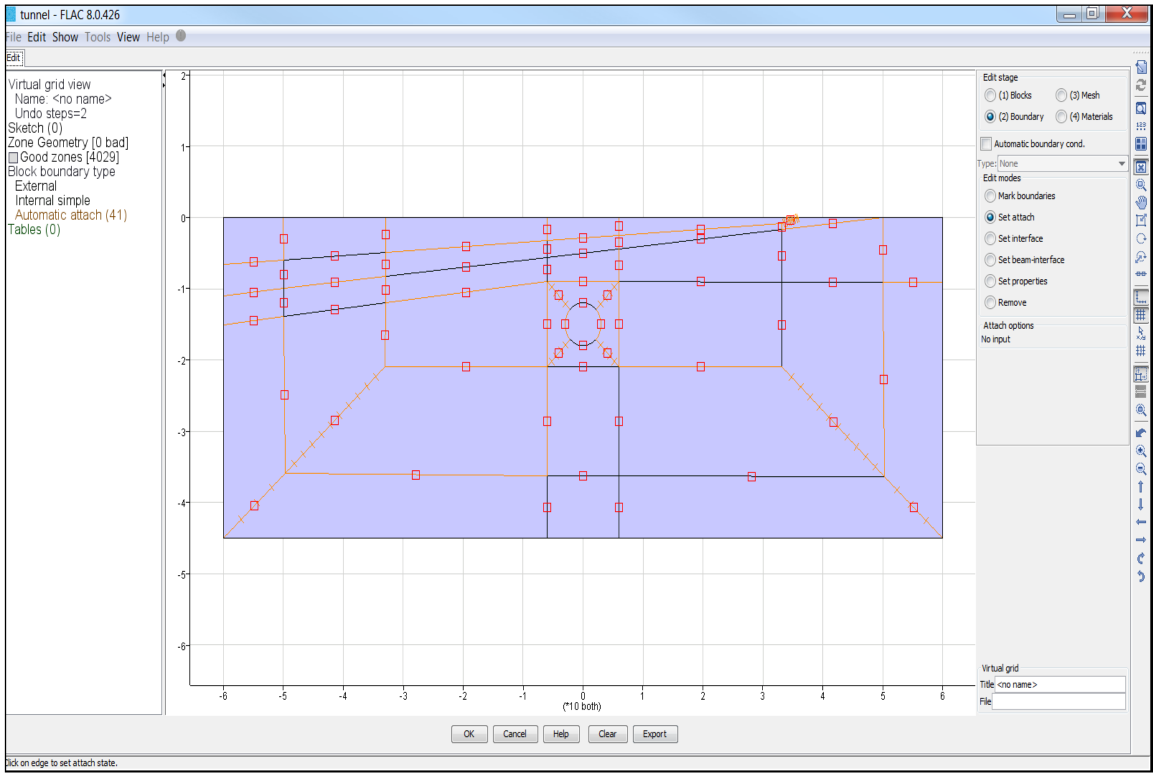
Edit stage: [✓ (1) Create] (2) Cleanup (3) Transform (4) Blocks
Edit modes: [✓ Add edges] Move points Edit edges Assign groups Delete points Delete edges
Edge group: Select construction line
Add edge options: Line Box Circle [✓ Horizontal] Vertical Arc
[✓] Intersect new edges to old
Circle points 32
Start angle 0
Arc length 180
File
Title <no name>

Figure 5.12 Add construction lines to divide the model into quadrilateral blocks.

A "sub-grid" is defined in Section 3.4.1.

Press [OK] to exit the [Geometry builder] tool, and then enter the [Edit] tool to generate and manipulate the zoning for the model. First select the [Boundary] edit mode in the [Edit] tool, as shown in Figure 5.13. If construction lines are shown in black in the [Edit] tool, this indicates the grid is continuous across the line. If the construction line is red, this indicates that sub-grids are attached at the construction line. In order to optimize zoning, it is recommended that the number of attached lines be reduced as much as possible in a model. This can be accomplished by using the [Set attach] edit mode to manually reset internal boundary lines as attached lines. For example, the tunnel periphery lines and the box lines around the tunnel are set as attached lines in Figure 5.14. Consequently, the number of attached lines is reduced from 46 to 17. Compare Figure 5.14 to Figure 5.13.

The [Mesh] edit stage is now selected to create and manipulate zoning within the model. Zoning can be set manually or automatically. Start by selecting "Use automatic zoning" and choosing "Fine 100" in the *Zoning options* dialog. This creates the mesh shown in Figure 5.15.

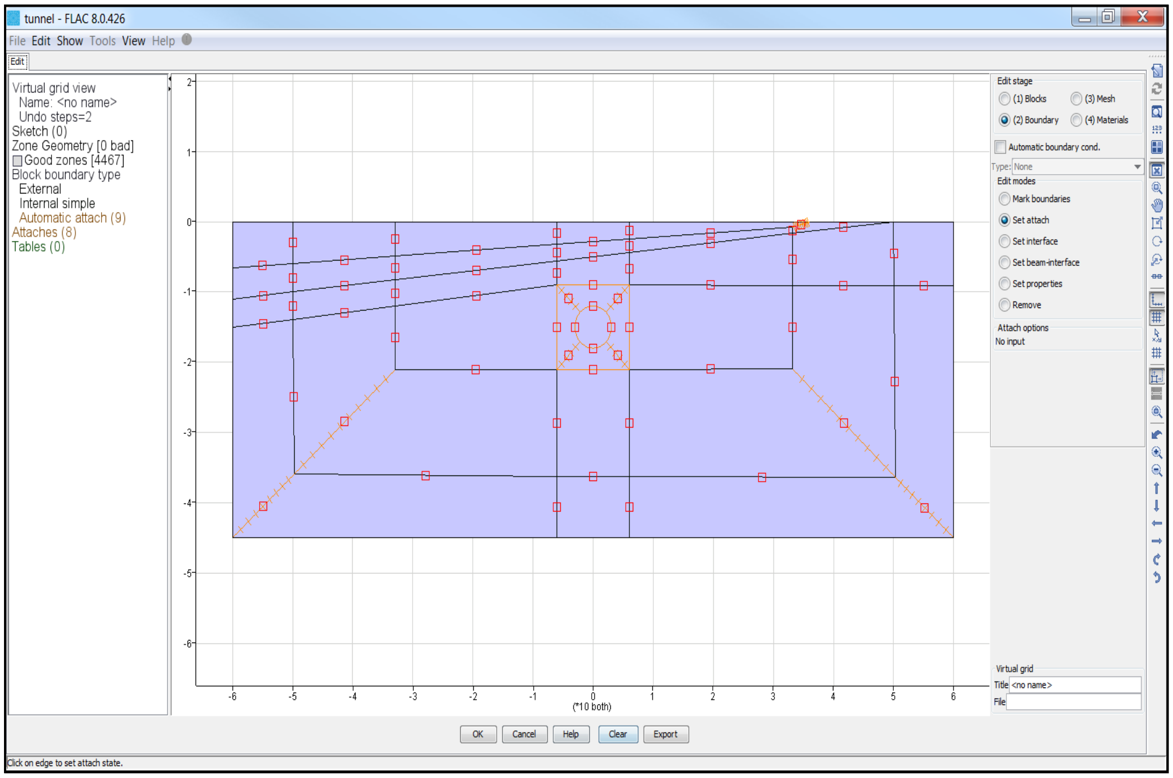


Virtual grid view
Name: <no name>
Undo steps=2
Sketch (0)
Zone Geometry [0 bad]
Good zones [4029]
Block boundary type
External
Internal simple
Automatic attach (41)
Tables (0)

Edit stage
(1) Blocks [✓ (2) Boundary] (3) Mesh (4) Materials
Automatic boundary cond.
Type: None
Edit modes
Mark boundaries [✓ Set attach] Set interface Set beam-interface Set properties Remove
Attach options
No input
Virtual grid
Title <no name>
File

lick on edge to set attach state.

Figure 5.13 Select the [Boundary] edit stage of the [Edit] tool.

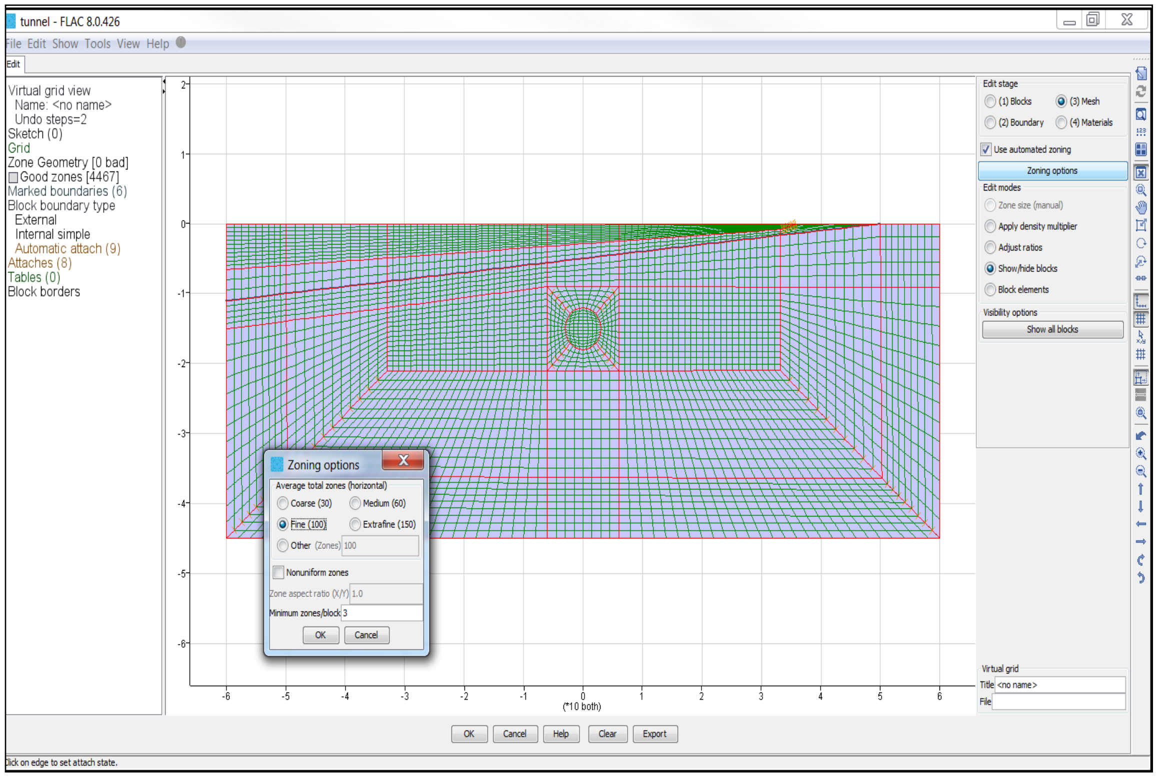


Virtual grid view
Name: <no name>
Undo steps=2
Sketch (0)
Zone Geometry [0 bad]
Good zones [4467]
Block boundary type
External
Internal simple
Automatic attach (9)
Attaches (8)
Tables (0)

Edit stage
(1) Blocks [✓ (2) Boundary] (3) Mesh (4) Materials
Automatic boundary cond.
Type: None
Edit modes
Mark boundaries [✓ Set attach] Set interface Set beam-interface Set properties Remove
Attach options
No input
Virtual grid
Title <no name>
File

Click on edge to set attach state.

Figure 5.14 Reduce the number of attached lines in the model by setting attached lines around the tunnel region.



Virtual grid view
Name: <no name>
Undo steps=2
Sketch (0)
Grid
Zone Geometry [0 bad]
Good zones [4467]
Marked boundaries (6)
Block boundary type
External
Internal simple
Automatic attach (9)
Attaches (8)
Tables (0)
Block borders

[✓ Use automated zoning]

Zone size (manual)
Apply density multiplier
Adjust ratios
[✓ Show/hide blocks]
Block elements

Visibility options

Title <no name>
File

Zoning options
Average total zones (horizontal)
Coarse (30) Medium (60) [✓ Fine (100)] Extrafine (150)
Other (Zones) 100
Nonuniform zones
Zone aspect ratio (X/Y) 1.0
Minimum zones/block 3

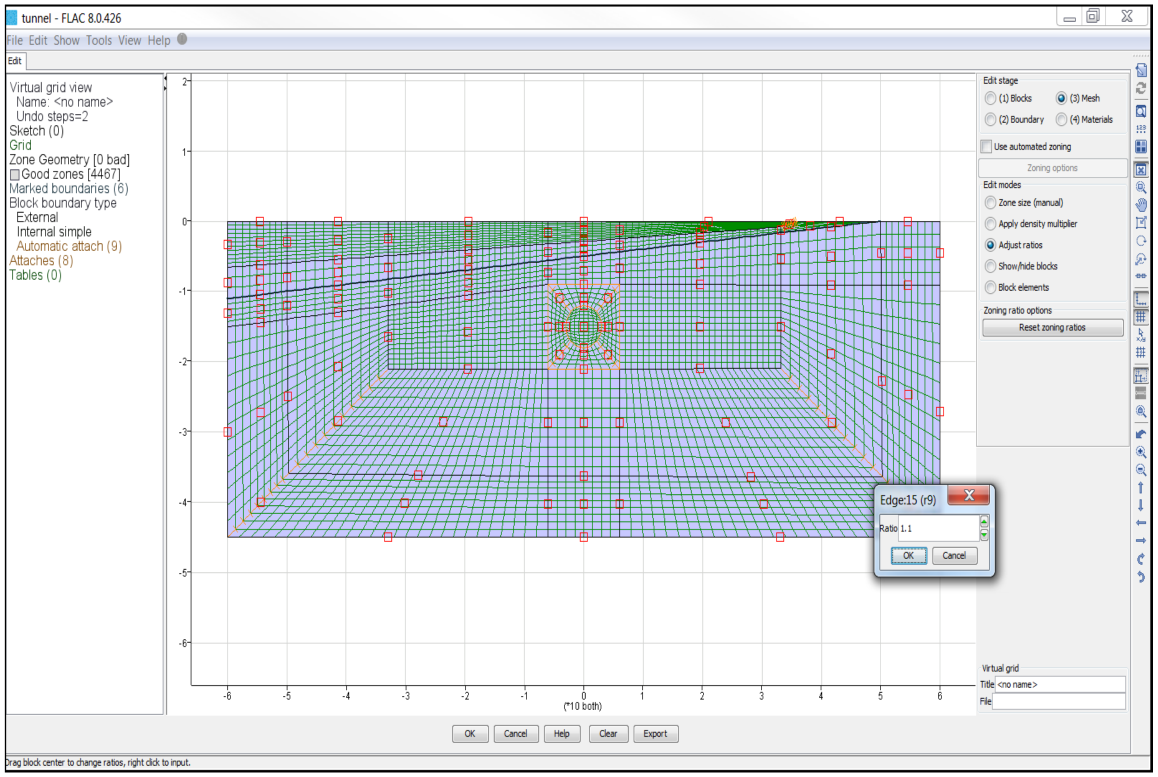
Click on edge to set attach state.

Figure 5.15 Create automatic zoning with the "Fine (100)" zoning option.

We consider this a "brute force" grid because the mesh is essentially uniform. The automatic zoning is now unchecked, so the grid can be adjusted manually. The number of zones along each side of the box of lines surrounding the tunnel is changed to 12, so the tunnel periphery will be composed of 48 segments. See Figure 5.16.

The number of zones at the outer boundary of the model is also reduced. By right-clicking on the red box in the outer ring of blocks at the model, the number of zones is changed from 9 to 4 (compare Figure 5.16 to Figure 5.15). The geometric ratio for zones in these blocks is set to 1.1 by selecting the "Adjust ratios" Edit mode and right-clicking on a red box to open the dialog as shown in Figure 5.16. The total number of zones is now reduced.

The zoning at this stage is considered appropriate for this problem; however, further refinement can still be made. It is always recommended to run models with different mesh densities to evaluate the effect of zone size on model results.


tunnel - FLAC 8.0.426

Virtual grid view
Name: <no name>
Undo steps=2
Sketch (0)
Grid
Zone Geometry [0 bad]
Good zones [4467]
Marked boundaries (6)
Block boundary type
External
Internal simple
Automatic attach (9)
Attaches (8)
Tables (0)

Edit stage: (1) Blocks [✓ (3) Mesh] (2) Boundary (4) Materials
Use automated zoning ☐
Edit modes: Zone size (manual) / Apply density multiplier / [✓ Adjust ratios] / Show/hide blocks / Block elements
Zoning ratio options: Reset zoning ratios

Edge:15 (r9)
Ratio 1.1

Virtual grid
Title <no name>
File

Drag block center to change ratios, right click to input.

Figure 5.16 Adjust zoning manually with [Zone size (manual)] and [Adjust ratios] Edit modes.

5.3.4 Boundary Conditions

The [Edit] tool also allows the specification of boundary conditions and assignment of material models and properties. The boundary conditions in a numerical model consist of the values of field variables that are prescribed at the boundary of the numerical grid. Boundaries can be either real or artificial—real boundaries exist in the physical object being modeled while artificial boundaries are introduced to enclose the chosen number of zones. Artificial boundaries fall into two categories: lines of symmetry and lines of truncation.[1] The two main types of mechanical conditions that can be applied at model boundaries are prescribed stress and prescribed displacement (velocity).

Experience has shown that, generally:

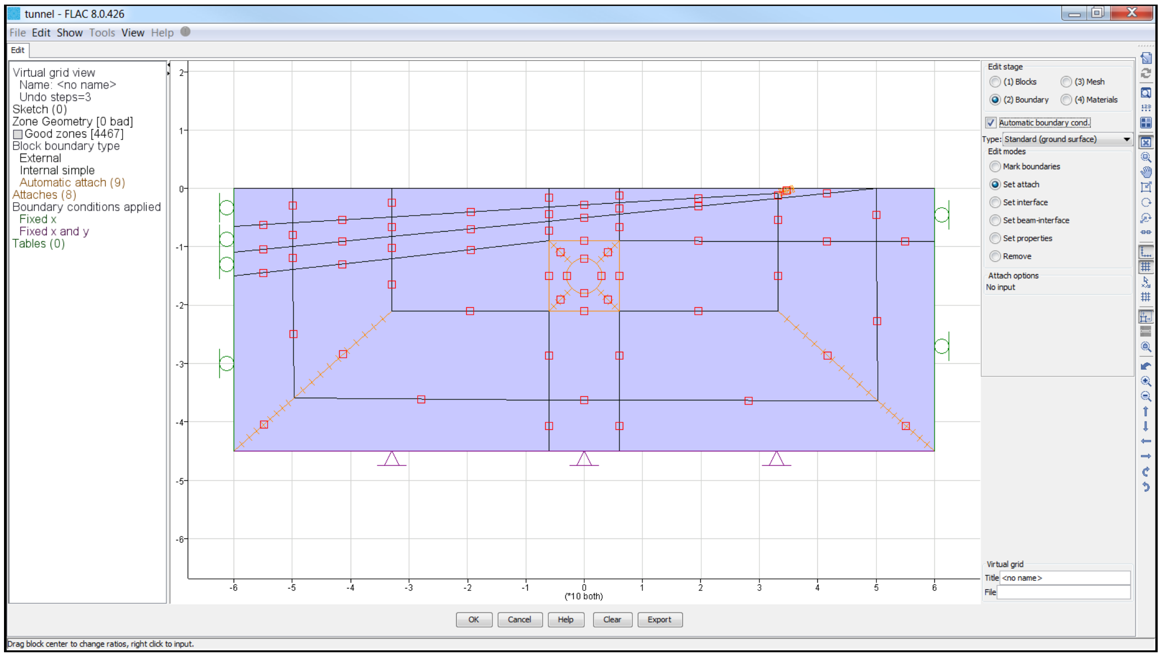
- a fixed zero velocity boundary (i.e., zero displacements) causes both stresses and displacements to be underestimated;
- a stress boundary causes both stresses and displacements to be overestimated; and
- the two types of boundary conditions bracket the true solution, so that it is possible to do tests with both boundaries and get a reasonable estimate of the true solution by averaging the two results.

The stress boundary will allow the sides of the model to deform so that the pressure equilibrium across the boundary can be maintained. Clearly, if the pressure applied is underestimated, the model would slump; if too much, it is crushed. Lithostatic values are not unreasonable as a starting point. Of course, the pressure build-up could be monitored and the boundary pressures increased to match, resulting in zero displacements.

1. **Artificial Boundaries—Lines of Symmetry:** Taking advantage of a line or axis of symmetry, which may be at any orientation, allows us to effectively halve the size of the physical model in its mathematical representation. Such a line is created by fixing velocities perpendicular to it, as there is no movement across it. It is not constrained in the parallel direction. **Artificial Boundaries—Lines of Truncation:** A model of a large body may be truncated at a boundary sufficiently far from the area of interest that the behavior in that area is not greatly affected. It helps to know how far away to place these boundaries and what errors might be expected in the stresses and displacements computed for the area of interest.

This is, in a sense, what happens at a zero displacement boundary. As discussed above, this is actually a *zero velocity* boundary. Forces are generated to match the forces tending to accelerate the gridpoints, leaving the gridpoint position unchanged. However, what is more concerning is that the top corners of the model try to move *inward* as the settlement trough develops. This may induce an artificial tensile failure as the fixed displacement condition causes the gridpoint to resist the desire to move to dissipate stress.

Fixed displacement boundaries are set in the [Boundary] edit stage in the [Edit] tool. Stress boundaries can be set in the [In Situ]/[Apply] tool. As shown in Figure 5.17, fixed, roller boundaries are set on the sides, and fixed *x*- and *y*-boundaries are set automatically on the base of the model if “Automatic boundary cond.” is selected. These are often called zero displacement boundary conditions, but in actuality, these are zero velocity boundary conditions in *FLAC*.



Virtual grid view
Name: <no name>
Undo steps=3
Sketch (0)
Zone Geometry [0 bad]
Good zones [4467]
Block boundary type
External
Internal simple
Automatic attach (9)
Attaches (8)
Boundary conditions applied
Fixed x
Fixed x and y
Tables (0)

Edit stage
( ) (1) Blocks ( ) (3) Mesh
[✓ (2) Boundary] ( ) (4) Materials
[✓ Automatic boundary cond.]
Type: Standard (ground surface)
Edit modes
( ) Mark boundaries
[✓ Set attach]
( ) Set interface
( ) Set beam-interface
( ) Set properties
( ) Remove
Attach options
No input

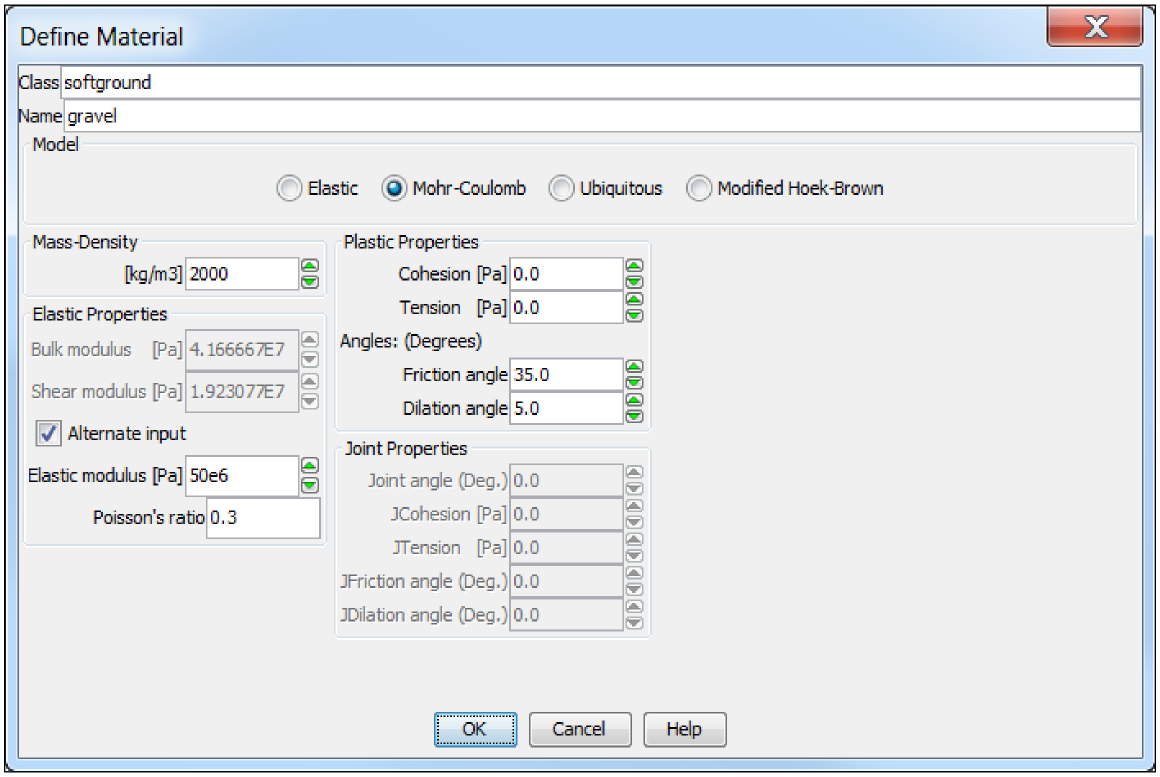
Virtual grid
Title <no name>
File

Drag block center to change ratios, right click to input.

Figure 5.17 Set boundary conditions in the [Boundary] edit stage of the [Edit] tool.

5.3.5 Materials Model and Properties

The Mohr-Coulomb material model is chosen to simulate the behavior of the clay and gravel materials in this example. This model is accessed using the [Materials] edit stage in the [Edit] tool. When [Create] is pressed, the *Define Material* dialog opens to assign the material name and properties, as shown for gravel in Figure 5.18.


Define Material

Class: softground
Name: gravel
Model: ( ) Elastic [✓ Mohr-Coulomb] ( ) Ubiquitous ( ) Modified Hoek-Brown

Mass-Density [kg/m3]: 2000

Elastic Properties
Bulk modulus [Pa]: 4.166667E7
Shear modulus [Pa]: 1.923077E7
[✓] Alternate input
Elastic modulus [Pa]: 50e6
Poisson's ratio: 0.3

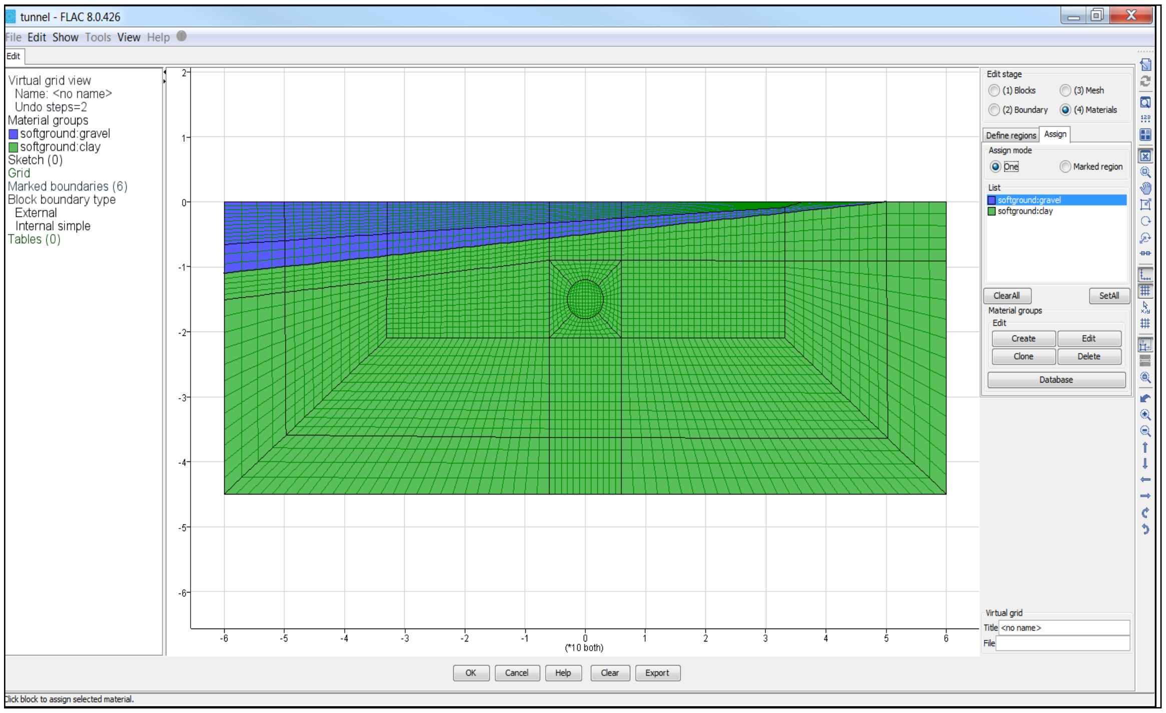
Plastic Properties
Cohesion [Pa]: 0.0
Tension [Pa]: 0.0
Angles: (Degrees)
Friction angle: 35.0
Dilation angle: 5.0

Joint Properties
Joint angle (Deg.): 0.0
JCohesion [Pa]: 0.0
JTension [Pa]: 0.0
JFriction angle (Deg.): 0.0
JDilation angle (Deg.): 0.0

Figure 5.18 Create materials and select properties in the *Define Material* dialog.

The gravel and clay materials are then assigned to blocks of zones by highlighting the material and clicking the mouse over the selected blocks. The material assignment appears as shown in Figure 5.19 after the gravel and clay materials are assigned.

The *FLAC* commands to create the model at this stage are generated by first pressing [OK] to exit the [Edit] tool, and then pressing [Execute] to send the commands to *FLAC*. The model state and associated *FLAC* commands are shown in Figure 5.20. The state is saved by pressing [Save]. The saved state is named “tun1.sav”, as shown in Figure 5.20.


tunnel - FLAC 8.0.426

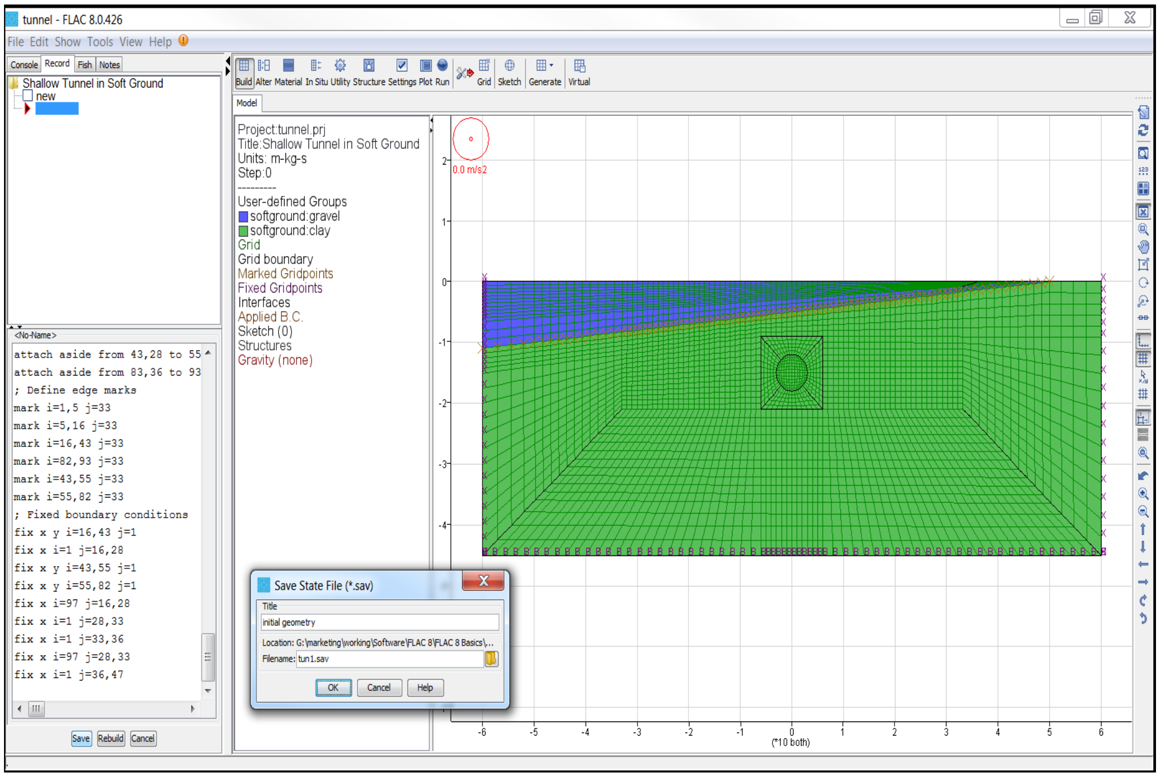
Virtual grid view
Name: <no name>
Undo steps=2
Material groups
🟦 softground:gravel
🟩 softground:clay
Sketch (0)
Grid
Marked boundaries (6)
Block boundary type
External
Internal simple
Tables (0)

Edit stage: ( ) (1) Blocks ( ) (2) Boundary ( ) (3) Mesh [✓ (4) Materials]
Assign mode: [✓ One] ( ) Marked region
List:
🟦 softground:gravel
🟩 softground:clay

Virtual grid
Title: <no name>
File:

Click block to assign selected material.

Figure 5.19 Assign gravel and clay materials and properties in the [Materials] edit stage.



```
attach aside from 43,28 to 55
attach aside from 83,36 to 93
; Define edge marks
mark i=1,5 j=33
mark i=5,16 j=33
mark i=16,43 j=33
mark i=82,93 j=33
mark i=43,55 j=33
mark i=55,82 j=33
; Fixed boundary conditions
fix x y i=16,43 j=1
fix x i=1 j=16,28
fix x y i=43,55 j=1
fix x y i=55,82 j=1
fix x i=97 j=16,28
fix x i=1 j=28,33
fix x i=1 j=33,36
fix x i=97 j=28,33
fix x i=1 j=36,47
```

Project:tunnel.prj
Title:Shallow Tunnel in Soft Ground
Units: m-kg-s
Step:0

Save State File (*.sav)
Title: initial geometry
Location: G:\marketing\working\Software\FLAC 8\FLAC 8 Basics\...
Filename: tun1.sav

Figure 5.20 Intial geometry of *FLAC* model.

5.3.6 Initial Conditions

In all civil or mining engineering projects, there is an in-situ state of stress in the ground, before any excavation or construction is started. By setting the initial conditions in the *FLAC* grid, an attempt is made to reproduce this in-situ state because it can influence the subsequent behavior of the model.

Ideally, information about the initial stress states comes from field measurements; the model, however, can be run for a range of possible conditions.

In a uniform layer of soil or rock with a free surface, the vertical stresses are usually equal to ρgh, where ρ is the mass-density of the material, g is the gravitational acceleration, and h is the depth below the surface. However, the in-situ horizontal stresses are more difficult to estimate. There is a common—but erroneous—belief that there is some "natural" ratio, K_0, between horizontal and vertical stress, given by $v/(1-v)$, where v is the Poisson's ratio. This formula is derived from the assumption that gravity is suddenly applied to an elastic mass of material in which lateral movement is prevented. This condition hardly ever applies in practice due to repeated tectonic movements, material failure, overburden removal, and locked-in stresses due to faulting and localization. Of course, if we had enough knowledge of the geological history of a particular volume of material, we might simulate the whole process numerically to derive the conditions immediately preceding our engineering works. This approach is often not feasible, so we compromise by installing a set of stresses in the grid and running the model until an equilibrium state is obtained.

It is important to realize that there is an infinite number of equilibrium states for any given system.

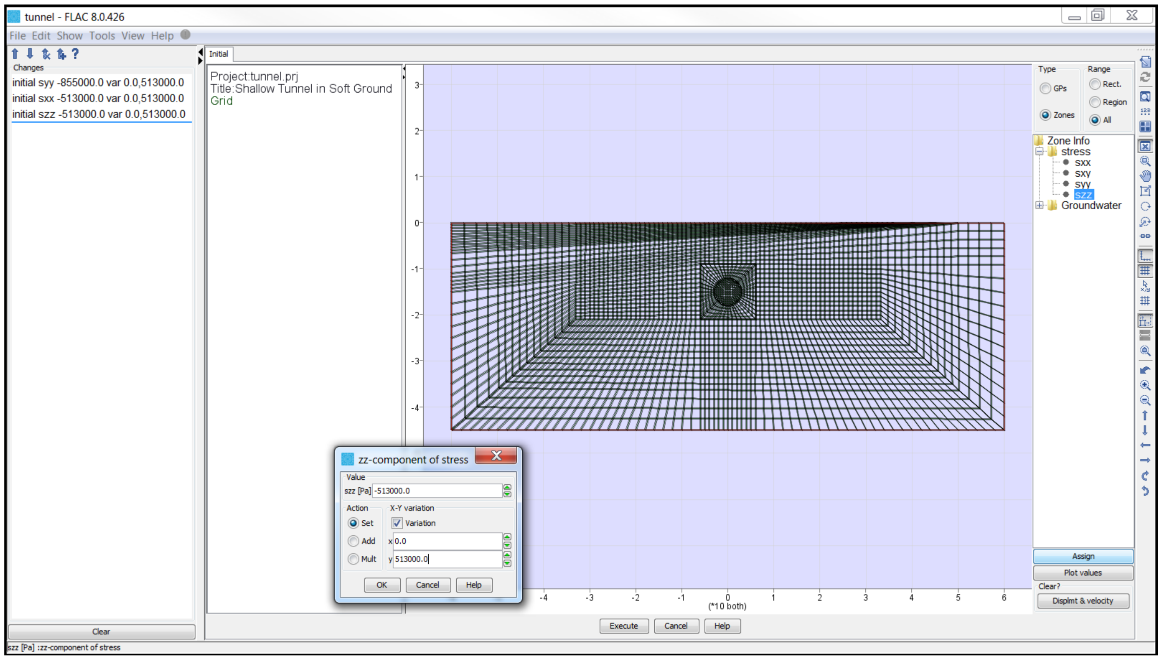
We could start with zero stresses in the model, but then we could not control K_0. Also, the model would take much longer to reach equilibrium.

The site investigation revealed a sloping horizon exiting at the surface. This made estimating the initial stresses difficult, but a ratio of horizontal stress to vertical stress K_0 equal to 0.6 is a good estimate. An initial state must be set up, letting the model reach equilibrium in such a way that the model K_0 is reasonably close to the desired K_0. The [In Situ]/[Initial] tool is used to initialize the stress state. See Figure 5.21. The vertical stress distribution is chosen to vary linearly from zero at the ground surface to 855,000 Pa at the base of the model, using the equation $\sigma_{yy} = \rho gh$, and approximating mass density throughout the model as 1900kg/m^3 and the gravitational acceleration loading as 10 m/sec^2. (The **INITIAL syy** command shown in Figure 5.21 applies this stress distribution.) The horizontal stress in the *x*-direction varies with depth by setting σ_{xx} and σ_{zz} equal to 0.6 * σ_{yy}. The idea is to set the model running from this state and let it find a suitable equilibrium state that takes into account the contrast in material properties and layered geometry.

Here linearly varying initial vertical and horizontal stresses are assigned to all zones in the model. (Note that "Range all" is selected in Figure 5.21.) The vertical stress at the top of the model is zero; at the bottom it is 0.855 MPa, acting in compression. Remember that stresses are associated with zone centroids and that negative values are compressive.

The linear variation in stress is assigned in the [In Situ]/[Initial] tool by checking "Variation" in each stress component dialog. A value is entered for variation in both the *x*- and *y*-direction. In this case, variation is only in the *y*-direction, so the *x*-value is zero. The *y*-value corresponds to the stress variation range starting from the lowest gridpoint position in the model (i.e., σ_{yy} = -0.855 MPa at the base) and ending at the highest point (i.e., σ_{yy} = 0 at the top). The variation is thus 0.855 MPa. Likewise, the variation for the horizontal stresses, σ_{xx} and σ_{zz} , is 0.513 MPa.

Gravity loading must also be specified. This is performed using the [Settings]/[Gravity] tool, which opens the Gravity settings dialog, as shown in Figure 5.22. The gravity loading applies body forces to all gridpoints in the model, and these forces correspond to the weight of the material surrounding each gridpoint. If the gravity loading is exactly balanced by the initialized stress distribution, then the model will be in equilibrium. However, an equilibrium calculation will usually be required to reduce a force imbalance.



```
initial syy -855000.0 var 0.0,513000.0
initial sxx -513000.0 var 0.0,513000.0
initial szz -513000.0 var 0.0,513000.0
```

Project:tunnel.prj
Title:Shallow Tunnel in Soft Ground
Grid

zz-component of stress

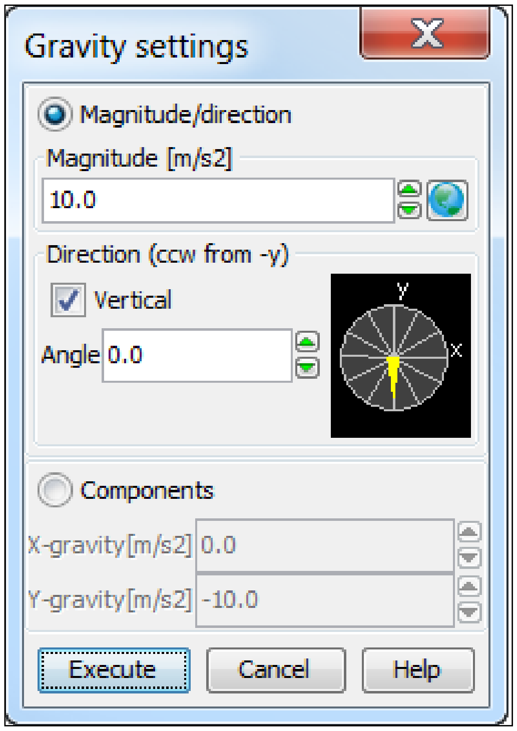
Value: szz [Pa] -513000.0

Action: [✓ Set] Add Mult

X-Y variation: [✓ Variation] x 0.0, y 513000.0

szz [Pa] :zz-component of stress

Figure 5.21 Stress distribution is initialized using the [In Situ]/[Initial] tool.


Gravity settings

Magnitude/direction [✓]

Magnitude [m/s2]: 10.0

Direction (ccw from -y)

Vertical [✓]

Angle 0.0

Components

X-gravity[m/s2] 0.0

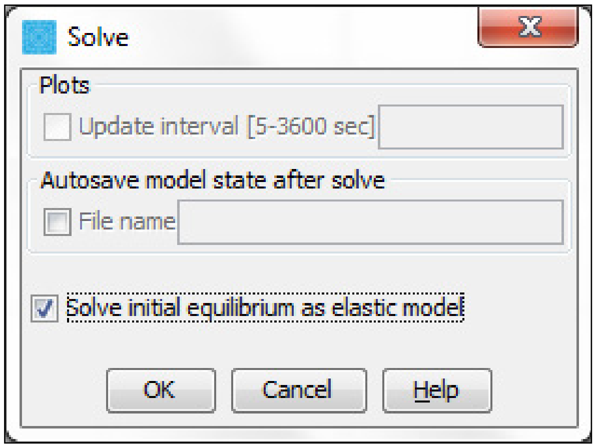
Y-gravity[m/s2] -10.0

Figure 5.22 Gravitational loading is assigned using the [Settings]/[Gravity] tool.

5.3.7 Running to Equilibrium

All the preparatory work is complete and the model can now be run to an equilibrium force state. This is accomplished using the [Run]/[Solve] tool, which opens the dialog shown in Figure 5.23. Gridpoint displacement will be calculated using the equations of motion, and kinetic energy will be dissipated as the model comes to equilibrium. Stresses at zone centroids can change and may experience a stress path that moves (temporarily) outside the material failure envelope. In this model, tension failure could occur at the exit area of the sloping horizon at the surface, simply because the stress points are initially very close to the failure envelope. To avoid this, exaggerated material strength properties can be applied while the model is reaching equilibrium, then reset to realistic values and the calculation continued to equilibrium. This is accomplished automatically by checking "Solve initial equilibrium as elastic model" in the *Solve* dialog, as shown in Figure 5.23

By default, the [Run]/[Solve] calculation stops when the ratio of maximum unbalanced force divided by the applied force at all gridpoints falls below 10^{-3}.


Solve

Plots

Update interval [5-3600 sec] [ ]

Autosave model state after solve

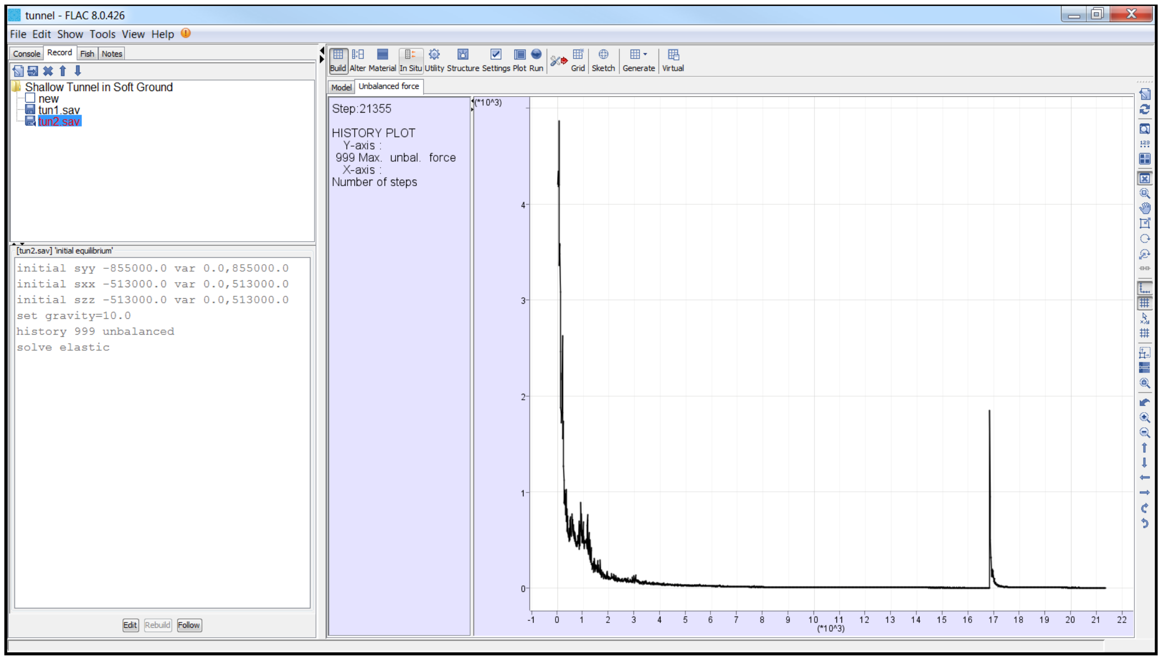
File name [ ]

Solve initial equilibrium as elastic model [✓]

Figure 5.23 Solve for initial equilibrium using the [Run]/[Solve] tool and check "Solve initial equilibrium as elastic model".

The model takes several thousand calculation steps to reach equilibrium. When the calculation stops, equilibrium is confirmed by plotting the maximum unbalanced force in the model using the [Plot]/[Quick] tool and selecting [unbalanced force]. The plot shown in Figure 5.24 shows that the maximum unbalanced force (and consequently kinetic energy) in the model has been reduced to an insignificant value. This indicates the model is in equilibrium. The spike near the end of the plot corresponds to the point in the calculation that the strength properties are reset to realistic values.

This state is saved by pressing [Save]. The saved state is named “tun2.sav” and is the state of the model when the tunnel construction is performed.


tunnel - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Build Alter Material In Situ Utility Structure Settings Plot Run
Grid Sketch Generate Virtual
Model Unbalanced force
Shallow Tunnel in Soft Ground
new
tun1.sav
tun2.sav
[tun2.sav] 'initial equilibrium'
Edit Rebuild Follow
(*10^3)

```
initial syy -855000.0 var 0.0,855000.0
initial sxx -513000.0 var 0.0,513000.0
initial szz -513000.0 var 0.0,513000.0
set gravity=10.0
history 999 unbalanced
solve elastic
```

```
Step:21355

HISTORY PLOT
   Y-axis :
 999 Max.  unbal.  force
   X-axis :
Number of steps
```

Figure 5.24 Plot maximum unbalanced force to check equilibrium using the [Plot]/[Quick] tool.

5.3.8 Modeling a Lined Tunnel

As discussed in Section 5.1, an important issue in the design of supports for tunnels is the amount of relaxation that takes place before installing the support. If no relaxation is allowed in the model, the loads acting on the support will be overpredicted because some relaxation actually takes place ahead of the excavation. If complete relaxation is allowed prior to the installation of the support, the load will be zero, given that equilibrium can be obtained without support.

In reality, some relaxation takes place, so it needs to be simulated. In this example, the **convergence-confinement method** is demonstrated to accomplish this.

5.3.9 Calculation of a Ground Reaction Curve in *FLAC*

The ground reaction curve can be calculated using the [In Situ]/[Apply] tool, which accesses the **APPLY relax** command to provide control over the rate of unloading of the tunnel boundary. The curve can be calculated directly in *FLAC* using the following approach.

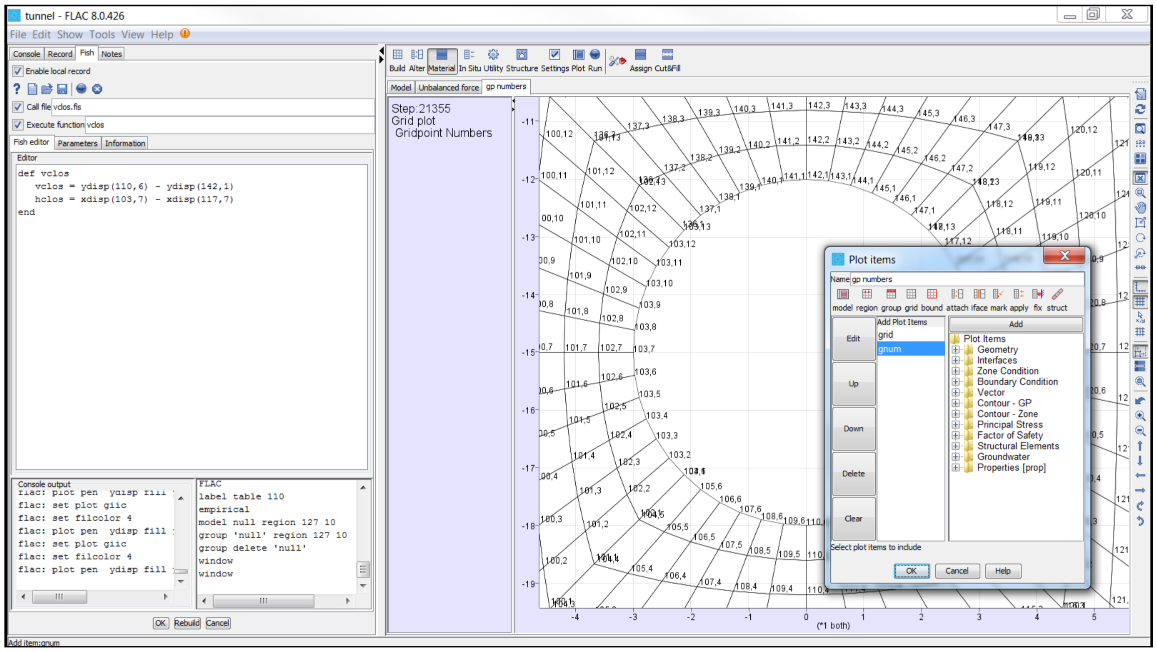
1. The gridpoint displacements in the model are initialized to zero. This will provide the starting point for developing the ground reaction curve. Gridpoint displacements are reset to zero by selecting [Clear? (Displmt & velocity)] in the [In Situ]/[Initial] tool.

2. The tunnel is excavated in the [Material]/[Assign] tool. The “Region” range is checked and “null” is selected from the material list. Click inside the tunnel region to null all tunnel zones.

3. Tunnel closure is monitored by recording displacement histories at selected gridpoints around the tunnel boundary. Gridpoints can be selected from a gridpoint ID number plot, generated in the [Plot]/[Model] tool, as shown in Figure 5.25. A simple *FISH* function is written to calculate the closure. In Figure 5.25, the *FISH* function **vclos** calculates the vertical closure of the tunnel as history **vclos** and the horizontal closure, **hclos**, in the *FISH editor* pane. (Note: The *FISH* function name must match a *FISH* history variable name in order for the histories to be recognized as history variables.) Execute the *FISH* function and press [OK] to return to the *Record* pane.

4. The *FISH* variables **vclos** and **hclos** are recorded as histories by selecting [*FISH* –> History] in the [Utility]/[History] tool.

5. In the [In Situ]/[Apply] tool, the [Force]/[Relax] boundary condition type is used to apply reaction forces as boundary forces around the tunnel. Select the “Long” Path in the tool and click on the tunnel boundary to designate the path, as shown in Figure 5.26. Press [Assign] to open the *Apply relax* dialog, as shown in the figure. The tunnel forces are relaxed in 20 relaxation steps to an end factor of 0.1, i.e., 90% relaxation. (If the force relaxation is 100%, the tunnel will collapse in this example.) Ground reaction curve tables are generated during the tunnel force relaxation when [Generate Ground Reaction Table] is selected. The vertical closure versus tunnel relaxation is selected to be written to Table 1, and the horizontal closure versus relaxation is written to Table 2.

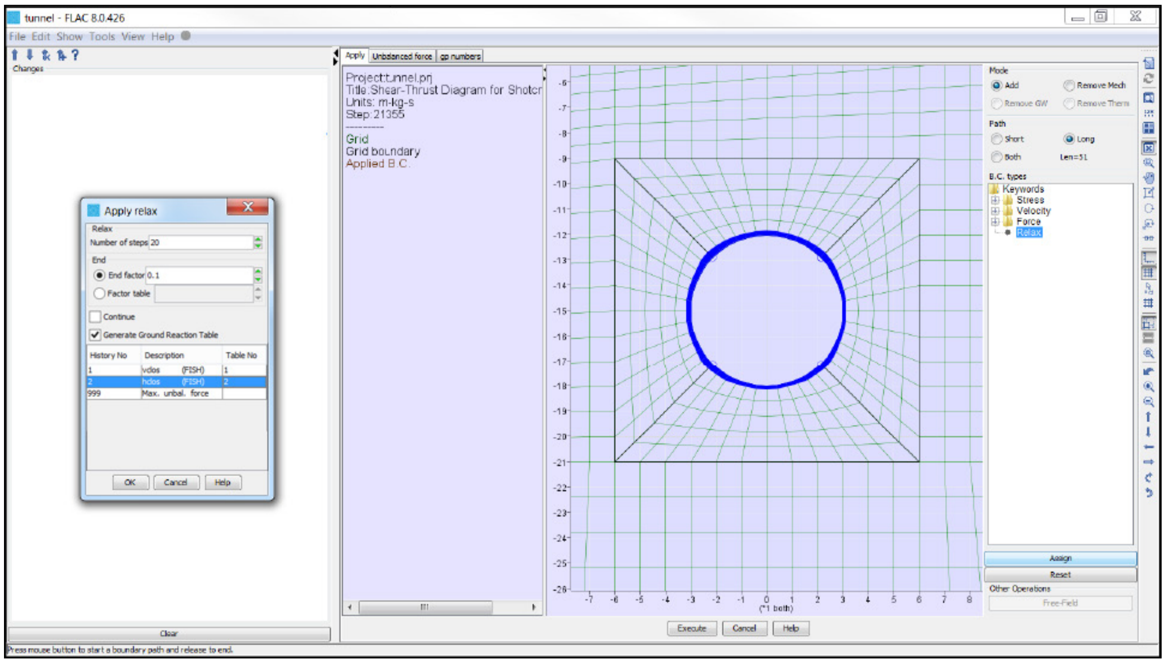


```
def vclos
   vclos = ydisp(110,6) - ydisp(142,1)
   hclos = xdisp(103,7) - xdisp(117,7)
end
```

```
flac: plot pen  ydisp fill
flac: set plot giic
flac: set filcolor 4
flac: plot pen  ydisp fill
flac: set plot giic
flac: set filcolor 4
flac: plot pen  ydisp fill
```

```
FLAC
label table 110
empirical
model null region 127 10
group 'null' region 127 10
group delete 'null'
window
window
```

Figure 5.25 **Plot gridpoint ID numbers in the [Plot]/[Model] tool and use the *FISH* editor to create *FISH* function vclos to create variables vlcos and hclos to monitor vertical and horizontal tunnel closure.**



Project:tunnel.prj
Title:Shear-Thrust Diagram for Shotcr
Units: m-kg-s
Step:21355

Grid
Grid boundary
Applied B.C.

Apply relax

Relax
Number of steps: 20

End
[✓ End factor] 0.1
Factor table

Continue
[✓] Generate Ground Reaction Table

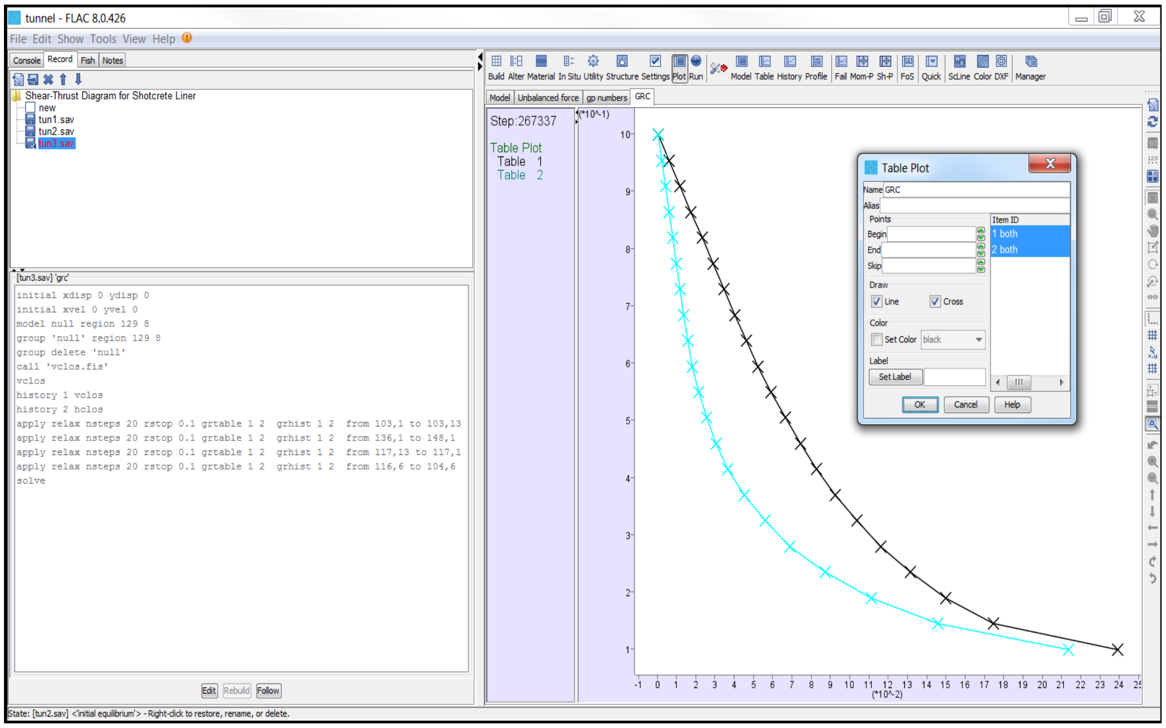
History No	Description	Table No
1	vclos (FISH)	1
2	hclos (FISH)	2
999	Max. unbal. force	

Press mouse button to start a boundary path and release to end.

Figure 5.26 **Apply reaction forces around the tunnel periphery in the [In Situ]/[Apply] tool.**

6. The effect of relaxation of tractions (i.e., stresses) around the tunnel is calculated using the [Run]/[Solve] tool. When the run stops, the ground reaction curve is plotted using the [Plot]/[Table] tool. Figure 5.27 shows the results. The relaxation factor, a ratio varying from 1.0 to 0.1, is plotted on the *y*-axis, and the closure (in meters) is plotted on the *x*-axis in this figure.

The model state at this stage is saved as "tun3.sav", as shown in Figure 5.27.



tunnel - FLAC 8.0.426
File Edit Show Tools View Help
Console
Record
Fish
Notes
Shear-Thrust Diagram for Shotcrete Liner
new
tun1.sav
tun2.sav
tun3.sav
Build Alter Material In Situ Utility Structure Settings Plot Run
Model Table History Profile Fail Mom-P Sh-P FoS Quick ScLine Color DXF Manager
Model
Unbalanced force
gp numbers
GRC
Step:267337
Table Plot
Table 1
Table 2
(*10^-1)
(*10^-2)
Edit
Rebuild
Follow

Table Plot

Name: GRC
Item ID: 1 both, 2 both
Draw: [✓] Line [✓] Cross
Set Color: black

```
[tun3.sav] 'grc'
initial xdisp 0 ydisp 0
initial xvel 0 yvel 0
model null region 129 8
group 'null' region 129 8
group delete 'null'
call 'vclos.fis'
vclos
history 1 vclos
history 2 hclos
apply relax nsteps 20 rstop 0.1 grtable 1 2  grhist 1 2  from 103,1 to 103,13
apply relax nsteps 20 rstop 0.1 grtable 1 2  grhist 1 2  from 136,1 to 148,1
apply relax nsteps 20 rstop 0.1 grtable 1 2  grhist 1 2  from 117,13 to 117,1
apply relax nsteps 20 rstop 0.1 grtable 1 2  grhist 1 2  from 116,6 to 104,6
solve
```

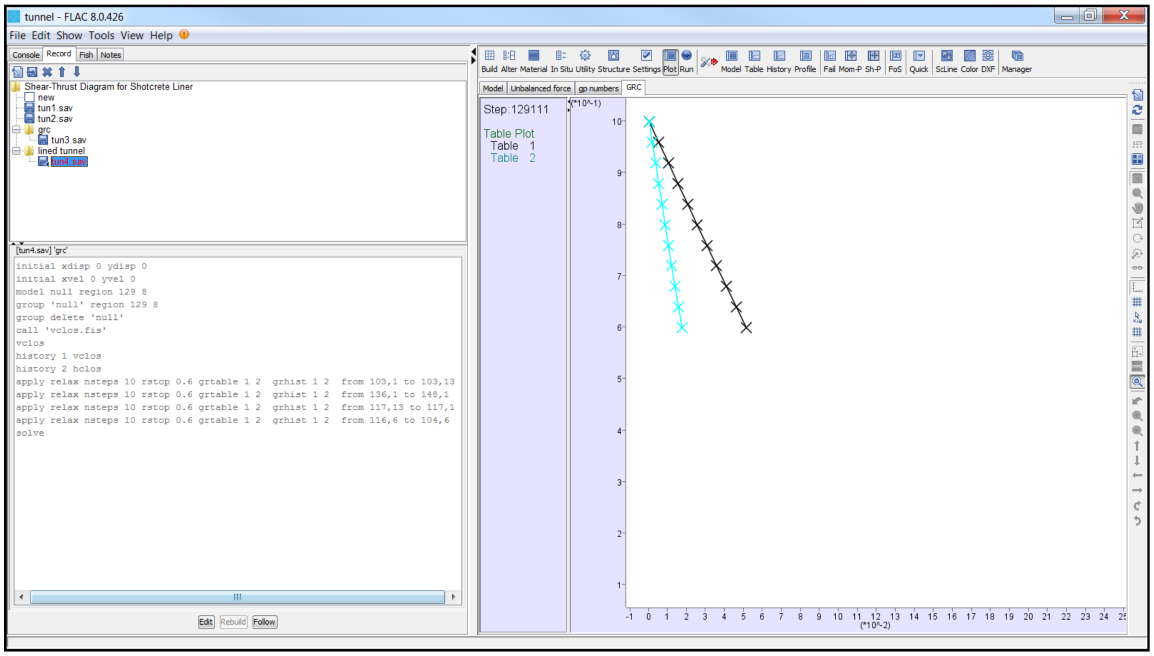
State: [tun2.sav] <'initial equilibrium'> - Right-click to restore, rename, or delete.

Figure 5.27 **Ground reaction curves showing normalized internal pressure in tunnel versus vertical and horizontal tunnel closure.**

5.3.10 Lining the Tunnel

It is now necessary to determine the amount of tunnel closure that occurs prior to installation of support. This typically is based upon the Longitudinal Displacement Profile (LDP), as discussed in Section 5.1, which can be obtained either from observed field values, empirical relations, or from numerical 3D models. In this exercise, it is assumed that, based upon field observation, approximately 5 cm vertical closure occurs before the liner is installed 2 m behind the tunnel face. Based upon the ground reaction curve, shown in Figure 5.27, this amount of closure can be related to an internal pressure relaxation of 40%.

The previous calculation state is now repeated, but this time the end factor is set to 0.6, i.e., 40% relaxation. This can easily be accomplished by cloning "tun3.sav". A state can be cloned by right-clicking on the state name in the project tree. The number of relaxation steps (**nsteps**) is changed to 10 and the end factor (**rstop**) is changed to 0.6 in the *Record* pane. This new state is rebuilt, and the ground reaction curve shows that the run has stopped at 40% relaxation, as shown by the ground reaction curve in Figure 5.28. The model state at this stage is saved as "tun4.sav". (Note that a separate branch is created in the project tree. Branch names are changed by right-clicking on the name.)


tunnel - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Build Alter Material In Situ Utility Structure Settings Plot Run Model Table History Profile Fail Mom-P Sh-P FoS Quick ScLine Color DXF Manager
Model Unbalanced force gp numbers GRC
Step:129111
Table Plot
Table 1
Table 2
(*10^-1)
(*10^-2)
Edit Rebuild Follow

Shear-Thrust Diagram for Shotcrete Liner
- new
- tun1.sav
- tun2.sav
- grc
  - tun3.sav
- lined tunnel
  - tun4.sav

[tun4.sav] 'grc'

```
initial xdisp 0 ydisp 0
initial xvel 0 yvel 0
model null region 129 8
group 'null' region 129 8
group delete 'null'
call 'vclos.fis'
vclos
history 1 vclos
history 2 hclos
apply relax nsteps 10 rstop 0.6 grtable 1 2  grhist 1 2  from 103,1 to 103,13
apply relax nsteps 10 rstop 0.6 grtable 1 2  grhist 1 2  from 136,1 to 148,1
apply relax nsteps 10 rstop 0.6 grtable 1 2  grhist 1 2  from 117,13 to 117,1
apply relax nsteps 10 rstop 0.6 grtable 1 2  grhist 1 2  from 116,6 to 104,6
solve
```

Figure 5.28 Ground reaction curves at 40% traction relaxation. ([Fix(Lock) View] in the View tool bar is pressed to lock the axes to match Figure 5.27.)

A partial shotcrete liner is now installed and placed from the tunnel knee (elevation is 16.15) to the crown. The liner is created in the [Structure]/[Liner] tool and is rigidly connected to the grid. See Figure 5.29. If the "Long" grid boundary path is selected, selected nodes can be erased by dragging the mouse over these nodes to create the partial liner geometry shown in Figure 5.29.

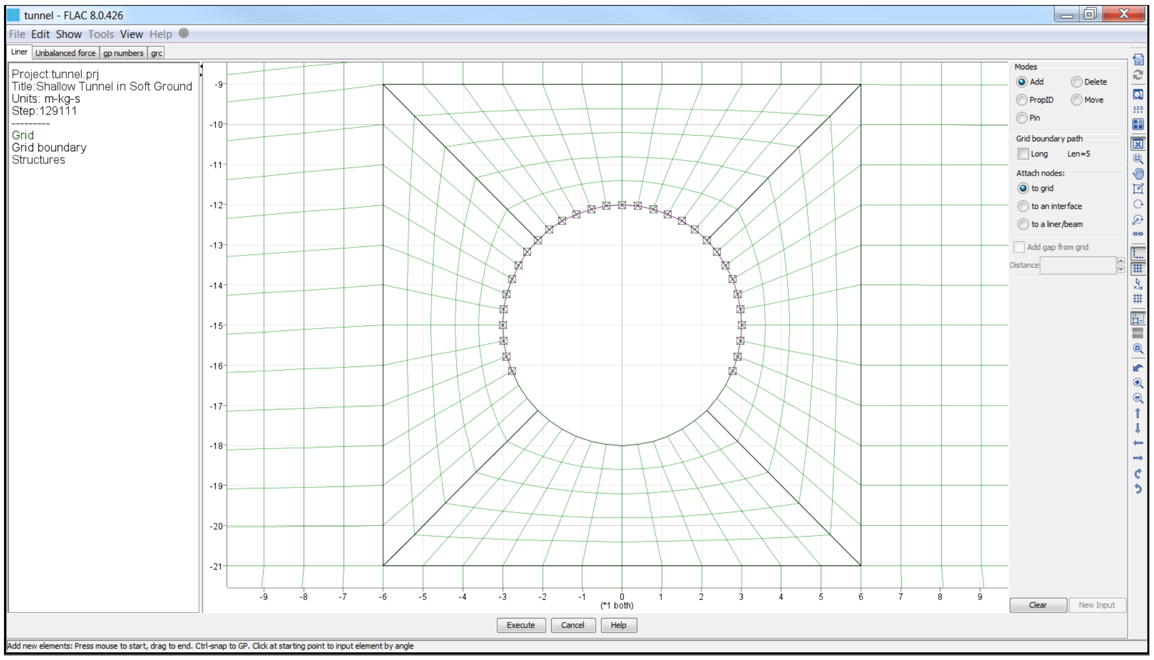
Liner properties are assigned in the [Structure]/[SEProp] tool. In this case, the effect of a partial shotcrete liner should be examined with

See Section 1.3.2 in the **Structural Elements** volume for a detailed description of liner properties.

Young's modulus = 5.5 GPa
Poisson's ratio = 0.2
shape factor = 0.83333
thickness = 0.3 m
cross-sectional area = 0.3 m^2
moment of inertia = 2.25x10^{-3} m^4
compressive strength = 10.0 MPa
tensile strength = 1.0 MPa

The tunnel tractions are now gradually reduced to zero using the [In Situ]/[Apply] tool. The forces are reduced from the 40% relaxation to zero in 10 steps. "Continue" is checked in the dialog, as shown in Figure 5.30, to indicate this is a continuation of the histories recorded for the previous relaxation.

The calculation is continued with the [Run]/[Solve] tool. The shotcrete liner placed from the tunnel knee to crown is expected to experience the axial forces and moments as shown in Figures 5.31 and 5.32.

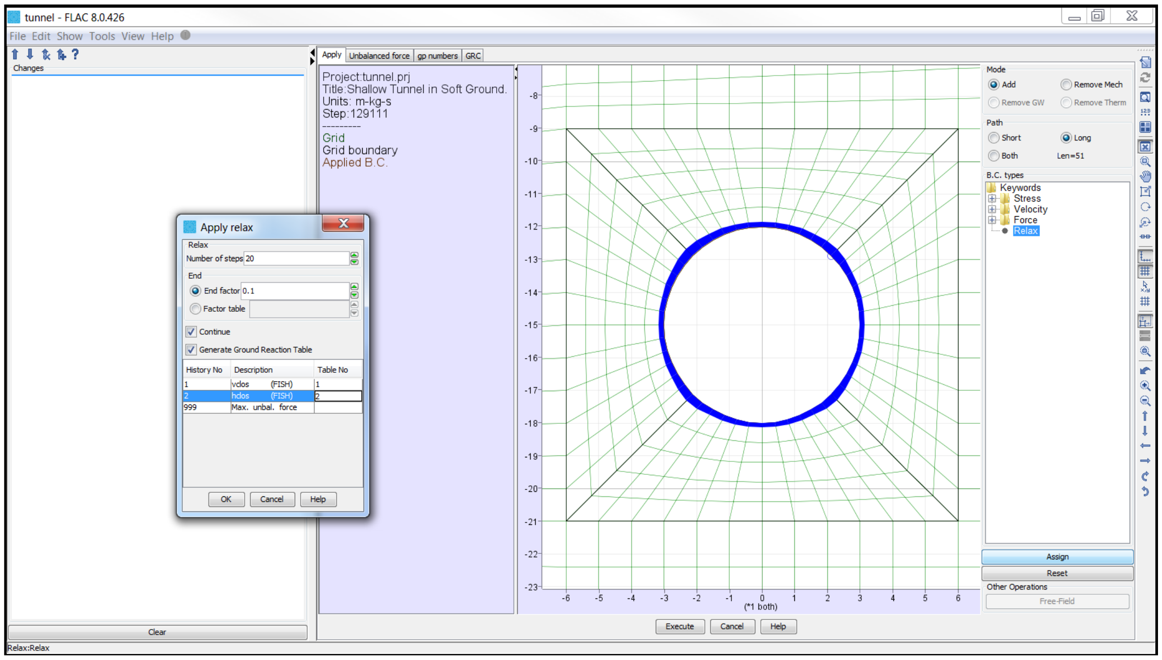


Project:tunnel.prj
Title:Shallow Tunnel in Soft Ground
Units: m-kg-s
Step:129111

Grid
Grid boundary
Structures

Add new elements: Press mouse to start, drag to end. Ctrl-snap to GP. Click at starting point to input element by angle

Figure 5.29 Connect liner elements to gridpoints along the tunnel periphery.



Project:tunnel.prj
Title:Shallow Tunnel in Soft Ground.
Units: m-kg-s
Step:129111

Grid
Grid boundary
Applied B.C.

Apply relax

Relax
Number of steps 20

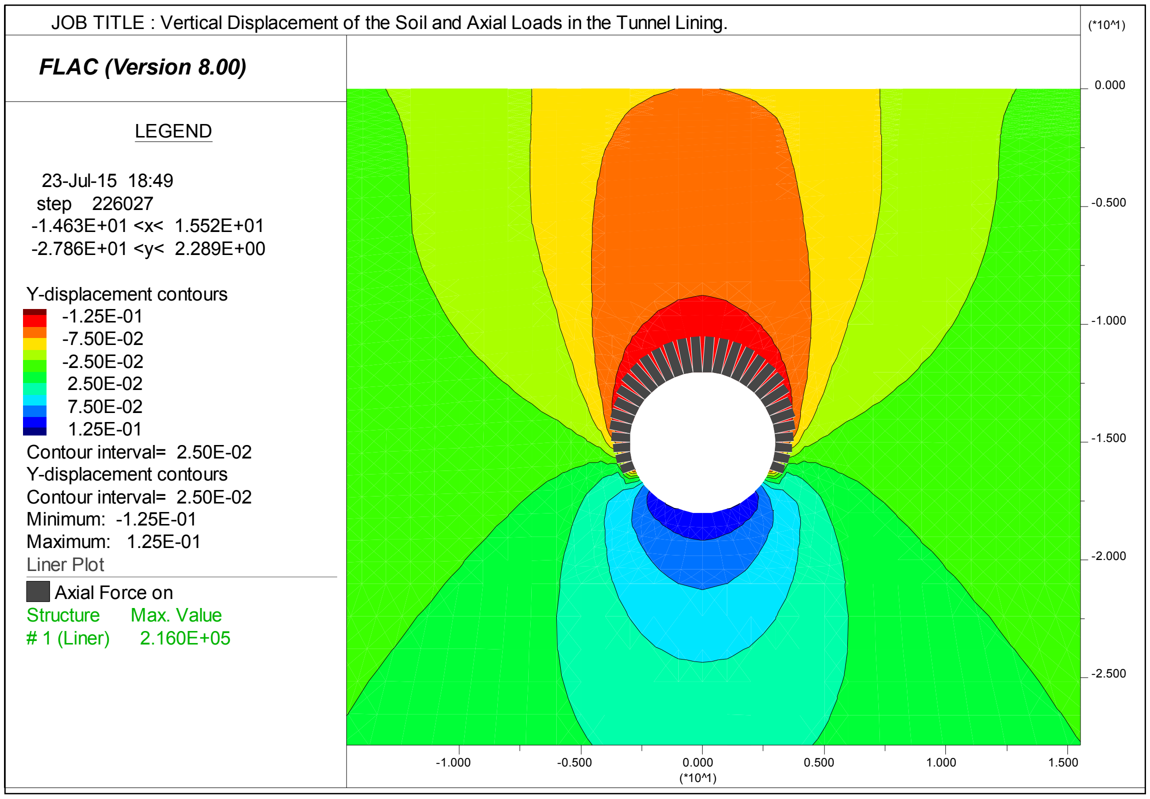
End
End factor 0.1 [✓]
Factor table

[✓] Continue
[✓] Generate Ground Reaction Table

History No	Description	Table No
1	vclos (FISH)	1
2	hclos (FISH)	2
999	Max. unbal. force	

OK Cancel Help

Figure 5.30 Continue relaxing forces around the tunnel periphery using the [In Situ]/[Apply] tool.



JOB TITLE : Vertical Displacement of the Soil and Axial Loads in the Tunnel Lining.

FLAC (Version 8.00)

LEGEND

23-Jul-15 18:49
step 226027
-1.463E+01 <x< 1.552E+01
-2.786E+01 <y< 2.289E+00

Y-displacement contours
-1.25E-01
-7.50E-02
-2.50E-02
2.50E-02
7.50E-02
1.25E-01
Contour interval= 2.50E-02
Y-displacement contours
Contour interval= 2.50E-02
Minimum: -1.25E-01
Maximum: 1.25E-01
Liner Plot

Axial Force on
Structure Max. Value
# 1 (Liner) 2.160E+05

Figure 5.31 Vertical displacement of the soil and axial loads in the tunnel lining.

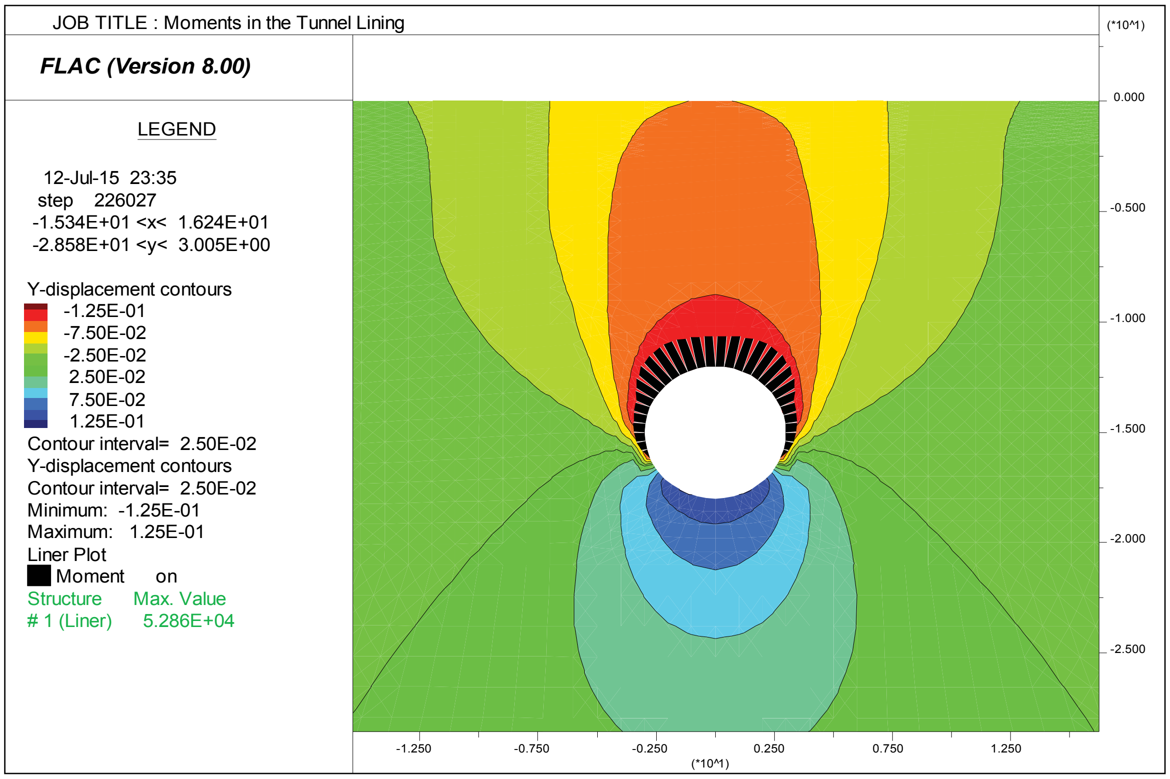


Figure 5.32 Vertical displacement of the soil and moments in the tunnel lining.

Support capacity diagrams can also be plotted to evaluate a factor of safety for the liner. Axial force versus moment and shear force versus moment envelopes are produced for given factors of safety of 1.0, 1.2, and 1.4. Values of axial force, moment, and shear force for the segments of the liner are plotted on the envelopes, as shown in Figures 5.33 and 5.34

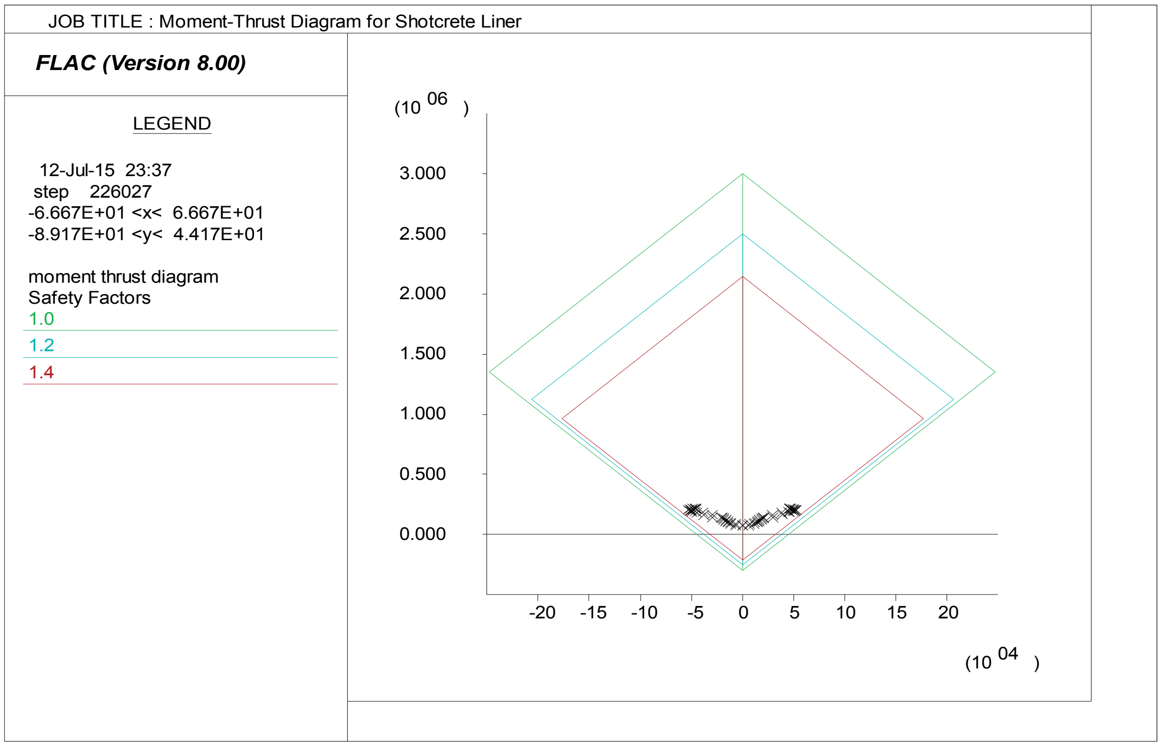


Figure 5.33 Moment-thrust diagram for shotcrete liner.

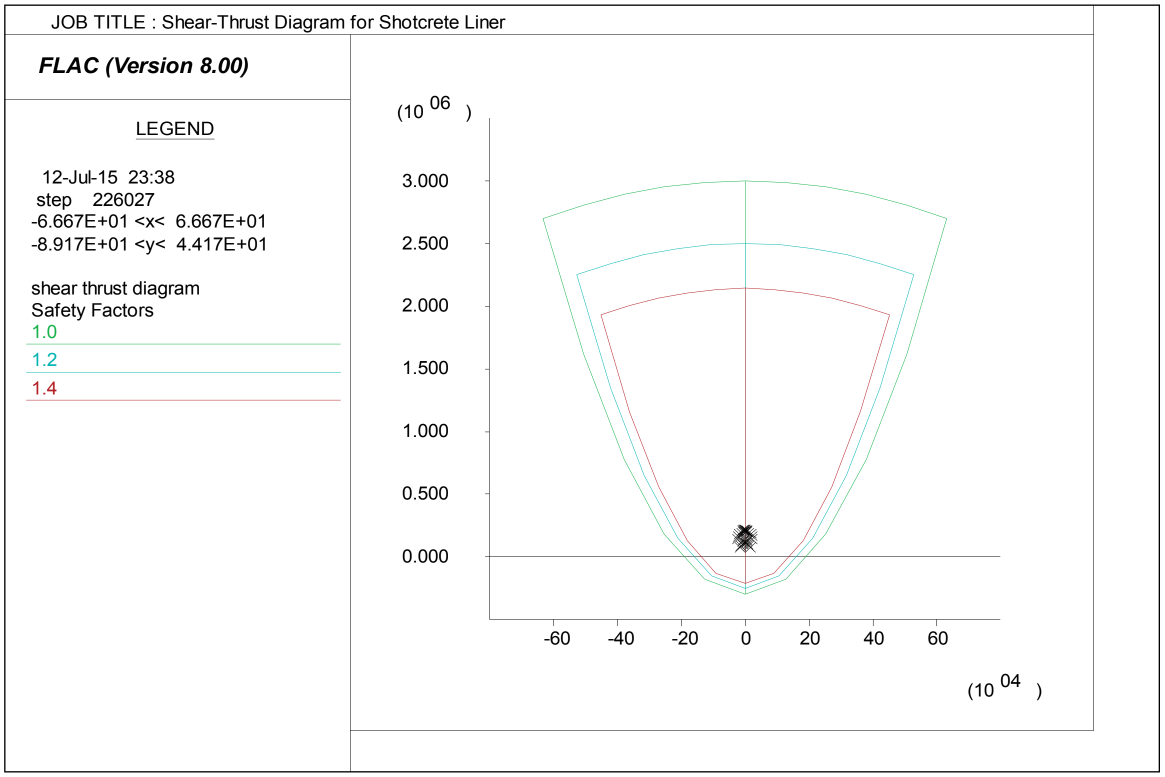


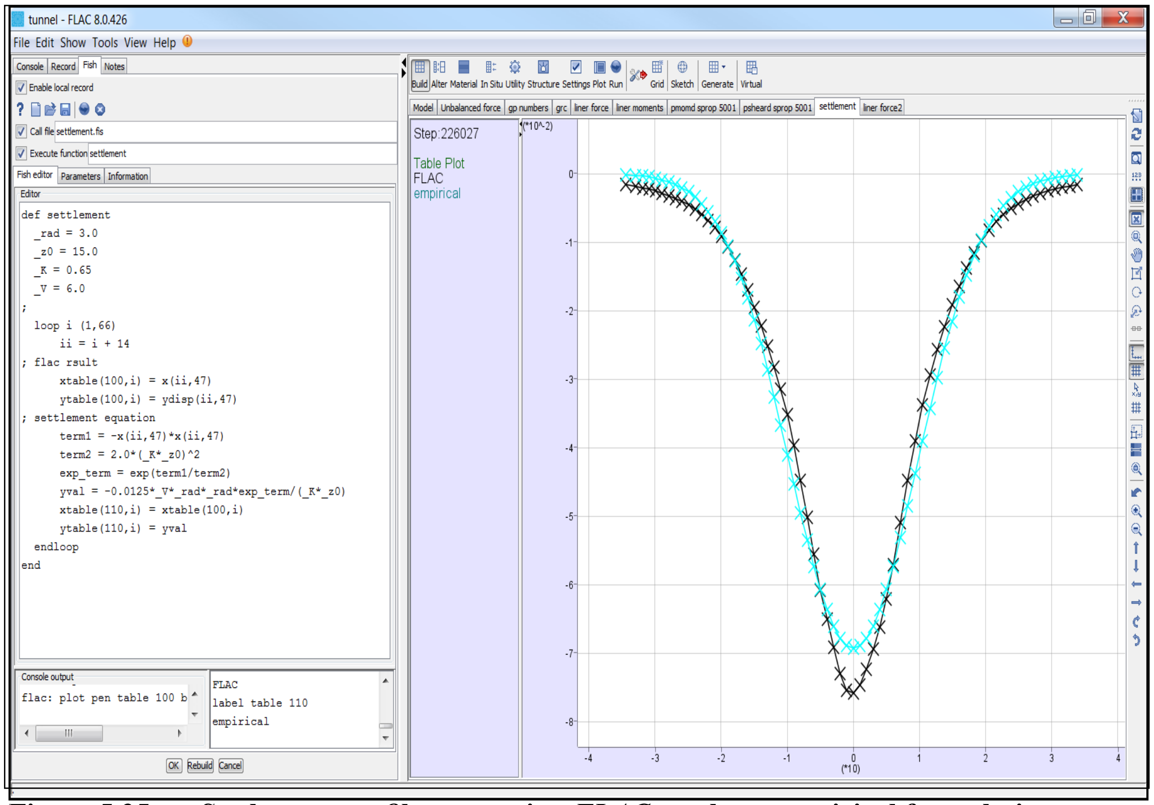
Figure 5.34 Shear-thrust diagram for shotcrete liner.

5.3.11 Surface Settlement Profile

The construction of this tunnel does produce a settlement trough at the ground surface, as shown in Figure 5.35. This plot is produced from the *FISH* function named **settlement** created in the *FISH* editor. The function stores the *y*-displacement values of the model surface gridpoints in table number 100. Note that surface gridpoint ID numbers are determined from the gridpoint ID plot in Figure 5.25. For this model, the settlement profile plotted at the ground surface ($y = 0$) extends from $x = -34.5$ m ($i = 15, j = 47$) to $x = 34.0$ m ($i = 80, j = 47$).

The *FISH* function also plots the empirical method equation from Section 5.1. The trough width parameter, *K*, is reported to be in the range 0.6 to 0.7, with a volume loss, *V*, between 2 and 10% for weak clays (O'Reilly and New, 1982[1]). Figure 5.35 shows a comparison between *FLAC* results and the empirical method selecting $K = 0.65$ and $V = 6\%$.

1. O'Reilly, M. P. and B. M. New "Settlements Above Tunnels in the United Kingdom, Their Magnitude and Prediction", in ***Proceedings of Tunneling '82*** (Brighton 1982), pp. 173–181 (1982).


tunnel - FLAC 8.0.426
File Edit Show Tools View Help
Console Record Fish Notes
Build Alter Material In Situ Utility Structure Settings Plot Run
Grid Sketch Generate Virtual
Model Unbalanced force gp numbers grc liner force liner moments pmomd sprop 5001 psheard sprop 5001 settlement liner force2
Step:226027
Table Plot
FLAC
empirical
(*10^-2)
(*10)
OK Rebuild Cancel

☑ Enable local record

☑ Call file settlement.fis

☑ Execute function settlement

Fish editor | Parameters | Information

Editor

```
def settlement
  _rad = 3.0
  _z0 = 15.0
  _K = 0.65
  _V = 6.0
;
  loop i (1,66)
      ii = i + 14
; flac rsult
      xtable(100,i) = x(ii,47)
      ytable(100,i) = ydisp(ii,47)
; settlement equation
      term1 = -x(ii,47)*x(ii,47)
      term2 = 2.0*(_K*_z0)^2
      exp_term = exp(term1/term2)
      yval = -0.0125*_V*_rad*_rad*exp_term/(_K*_z0)
      xtable(110,i) = xtable(100,i)
      ytable(110,i) = yval
  endloop
end
```

Console output

```
flac: plot pen table 100 b
```

```
FLAC
label table 110
empirical
```

Figure 5.35 Settlement profile comparing *FLAC* results to empirical formulation.

Remember that the empirical results were derived from observations of tunnels in overconsolidated clays. There are some data on soft clays (as referenced earlier), but the formulation is based overwhelmingly on stiffer material. While there may be debate on the nature of the deformation leading to the settlement shown in the *FLAC* model compared with the referenced empirical method, it is at least possible in the numerical analysis to examine the material behavior closely. One should, on principle, carry out a series of calculations to evaluate the influence of the different components of this problem.

5.4 On Your Own

1. The partial lining used in this example allows the tunnel invert to heave. How would the model react with a fully lined tunnel? How reasonable are these scenarios?

2. What is the effect of different tunnel shapes? Consider a horseshoe-shaped tunnel, for example.

3. Consider different types of structural support. For example, what if rockbolt support is installed first, corresponding to a relaxation of 20%, followed by shotcrete liner support at 50% relaxation?

Appendix A: Conventions

The following sign conventions are used in *FLAC* and must be kept in mind when entering input or evaluating results.

DIRECT STRESS	Positive stresses indicate tension; negative stresses indicate compression.
SHEAR STRESS	A positive shear stress points in the positive direction of the coordinate axis of the second subscript if it acts on a surface with an outward normal in the positive direction of the axis of the first subscript. Conversely, if the outward normal of the surface is in the negative direction, then the positive shear stress points in the negative direction of the coordinate axsis of the second subscript. The shear stresses shown in Figure A.1 all are positive.
DIRECT STRAIN	Positive strain indicates extension; negative indicates compression.
SHEAR STRAIN	Shear strain follows the same convention as shear stress. The distortion associated with positive and negative strain is illustrated in Figure A.1.
PRESSURE	A positive pressure will act normal to, and in a direction toward, the surface of a body (i.e., push). A negative pressure will act normal to, and in a direction away from, the surface of a body (i.e., pull).
PORE PRESSURE	Fluid pore pressure is positive in compression. Negative fluid pressure indicates fluid tension.
GRAVITY	Positive gravity will pull the mass of a body downward. Negative gravity will pull the mass of a body upward.
VECTOR QUANTITIES	The *x*- and *y*-components of vector quantities such as forces, displacements, velocities, and fluid flow are positive when pointing in the directions of the positive *x*- and *y*-coordinate space.

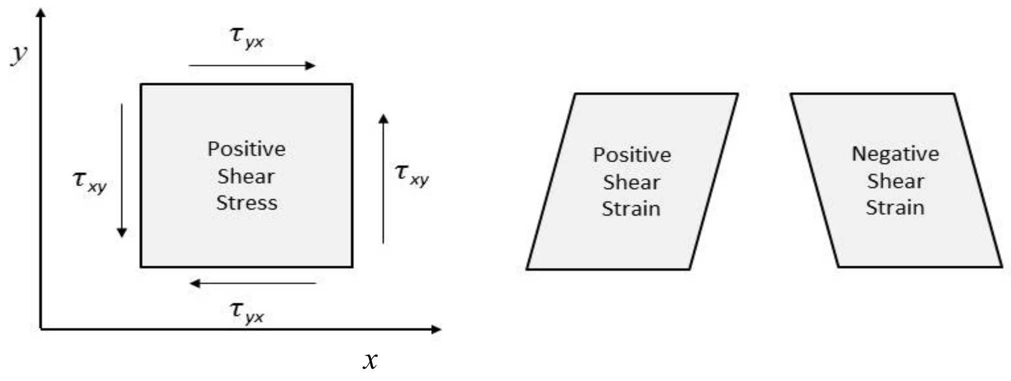


Figure A.1 Sign convention for shear stress and shear strain.

Appendix B: Systems of Units

FLAC accepts any consistent set of engineering units. Examples of consistent sets of units for basic parameters are shown in Tables B.1, B.2, and B.3. The user should exercise great care when converting from one system of units to another. No conversions are performed in *FLAC* except for friction and dilation parameters, which are converted from degrees to coefficients.

Table B.1 Systems of Units: Mechanical Properties

Property	SI				Imperial	
Length	m	m	m	cm	ft	in
Density	kg/m^3	10^3 kg/m^3	10^6 kg/m^3	10^6 g/cm^3	slugs/ft^3	snails/in^3
Force	N	kN	MN	Mdynes	lbf	lbf
Stress	Pa	kPa	MPa	bar	lbf/ft^2	psi
Gravity	m/s^2	m/s^2	m/s^2	cm/s^2	ft/s^2	in/s^2
Stiffness	Pa/m	kPa/m	MPa/m	bar/cm	lbf/ft^3	lb/in^3

Table B.2 Systems of Units: Groundwater Flow Parameters

Property	SI		Imperial	
Water Bulk Modulus	Pa	bar	lbf/ft^2	psi
Water Density	kg/m^3	10^6 g/cm^3	slugs/ft^3	snails/in^3
Permeability	m^3 s/kg	10^{-6} cm s/g	ft^3 s/slug	in^3 s/snail
Intrinsic Permeability	m^2	cm^2	ft^2	in^2
Hydraulic Conductivity	m/s	cm/s	ft/s	in/s

Table B.3 Systems of Units: Structural Elements

Property	Unit	SI				Imperial	
Area	length^2	m^2	m^2	m^2	cm^2	ft^2	in^2
Young's Modulus	stress	Pa	kPa	MPa	bar	lbf/ft^2	psi
Moment of Inertia	length^4	m^4	m^4	m^4	cm^4	ft^4	in^4
Bond Stiffness	force/len/disp	N/m/m	kN/m/m	MN/m/m	Mdynes/cm/cm	lbf/ft/ft	lbf/in/in
Axial/Shear Stiffness	force/disp	N/m	kN/m	MN/m	Mdynes/cm	lbf/ft	lbf/in
Plastic Moment	force length	N m	kN m	MN m	Mdynes cm	ft lbf	in lbf
Bond Strength	force/length	N/m	kN/m	MN/m	Mdynes/cm	lbf/ft	lbf/in
Yield Strength	force	N	kn	MN	Mdynes	lbf	lbf
Exposed Perimeter	length	m	m	m	cm	ft	in

ITASCA
Consulting Group, Inc.
111 Third Avenue South, Suite 450
Minneapolis, MN 55401 USA
phone: 612 371-4711
fax: 612 371-4717
email: software@itascacg.com
web: www.itascacg.com