NUREG/CR-5430
Vol. 2

FLAC (Fast Lagrangian Analysis of Continua) Version 2.20

User's Manual

Prepared by Mark Board

Itasca Consulting Group, Inc.

Prepared for
U.S. Nuclear Regulatory Commission

AVAILABILITY NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 2120 L Street, NW, Lower Level, Washington, DC 20555

2. The Superintendent of Documents, U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20013-7082

3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Office of Information Resources Management, Distribution Section, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

DISCLAIMER NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability of responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

NUREG/CR-5430
Vol. 2

FLAC (Fast Lagrangian Analysis of Continua) Version 2.20

User's Manual

Manuscript Completed: September 1989
Date Published: October 1989

Prepared by
Mark Board

Itasca Consulting Group, Inc.
1313 5th Street SE, Suite 210
Minneapolis, MN 55414

Prepared for
Division of High-Level Waste Management
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN D1016

ABSTRACT

FLAC (Fast Lagrangian Analysis of Continua), Version 2.20, is a two-dimensional, large-strain, explicit finite difference code written for analysis of problems in geotechnical engineering. FLAC has the ability to perform static mechanical analyses as well as transient heat transfer and fluid flow simulations. Various constitutive models are available to describe linear and non-linear response of the solid. Coupling can be performed between the thermal and mechanical, as well as the fluid and mechanical, models. The following report presents the documentation specified in NUREG-0856, Documentation of Computer Codes for High Level Waste Management. The documentation is presented in three volumes. Volume 1 contains the mathematical basis for the various aspects of the code; Volume 2 is the code User's Manual, and Volume 3 presents FLAC verification, example and benchmark problems.

ITASCA CONSULTING GROUP, INC.

TERMS AND CONDITIONS

FOR LICENSING FLAC

You should carefully read the following terms and conditions before using the FLAC program. Insertion of the FLAC disk into your computer indicates your acceptance of these terms and conditions. If you do not agree with them, you should promptly return the package and your money will be refunded.

This program is provided by Itasca Consulting Group, Inc. Title to the media on which the program is recorded and to the documentation in support thereof is transferred to the customer, but title to the program is retained by Itasca. You assume responsibility for the selection of the program to achieve your intended results and for the installation, use and results obtained from the program.

LICENSE

You may use the program on only one machine at any one time.

You may copy the program for back-up only in support of such use.

You may <u>not</u> use, copy, modify, or transfer the program, or any copy, in whole or part, except as expressly provided in this document.

You may <u>not</u> sub-license, rent, or lease this program.

FLAC
Terms and Conditions
Page (b)

TERM

The license is effective until terminated. You may terminate it any time by destroying the program together with the back-up copy. It will also terminate if you fail to comply with any term or condition of this agreement. You agree upon such termination to destroy the program together with the back-up copy, modifications, and/ or merged portions in any form.

WARRANTY

Itasca will correct any errors in the code at no charge for twelve (12) months after the purchase date of the code. Notification of a suspected error must be made in writing, with a complete listing of the input and output files and description of the error. If, in the judgment of Itasca, the code does contain an error, Itasca will (at its option) correct or replace the copy at no cost to the user or refund the initial purchase price of the code.

LIMITATION OF LIABILITY

Itasca assumes no liability whatsoever with respect to any use of FLAC or any portion thereof or with respect to any damages or losses which may result from such use, including (without limitation) loss of time, money or goodwill which may arise from the use of FLAC (including any modifications or updates that may follow). In no event shall Itasca be responsible for any indirect, special, incidental or consequential damages arising from use of FLAC.

CODE SUPPORT

Itasca will provide telephone support, at no cost, to assist the code owner in the installation of the FLAC code on his or her computer system. Additionally, general assistance may be provided in aiding the owner in understanding the capabilities of the various features of the code. However, no-cost assistance is not provided for help in applying FLAC to specific user-defined problems.

Technical support can be purchased on an as-needed basis for $70.00/hour in minimum one-half hour increments. For users who envisage the need for substantial amounts of assistance, support programs offering up to eight (8) hours of support in a twelve (12) month period for $500, or up to twenty (20) hours of support in a twelve (12) month period for $1,000 are available. In all instances, the user in encouraged to mail or fax the problem description to Itasca in order to minimize the amount of time spent trying to define the problem.

F L A C

Fast Lagrangian Analysis of Continua

(Version 2.2)

TABLE OF CONTENTS

F L A C
Table of Contents – 2

F L A C
Table of Contents - 3

F L A C
Table of Contents — 4

F L A C
Table of Contents — 5

F L A C
Table of Contents – 6

F L A C
Table of Contents - 7

F L A C
Table of Contents - 8

F L A C
Table of Contents - 9

F L A C
Table of Contents - 10

F L A C
Table of Contents – 11

LIST OF FIGURES

LIST OF FIGURES
(continued)

LIST OF FIGURES
(continued)

LIST OF FIGURES
(continued)

LIST OF FIGURES
(continued)

LIST OF TABLES

FLAC

Fast Lagrangian Analysis of Continua

(Version 2.2)

June 1989

1.0 INTRODUCTION

FLAC is an explicit finite difference code which simulates the behavior of structures built of soil, rock, or other materials which may undergo plastic flow when their yield limit is reached. Materials are represented by zones, or elements, which form a grid that is adjusted by the user to fit the shape of the object to be modeled. Each element follows a prescribed linear or non-linear stress/strain law in response to the applied forces and boundary restraints. If stresses are high enough to cause the material to yield and flow, the grid actually deforms and moves with the material it represents. This calculation scheme is called "Lagrangian" and is well suited to modeling large distortions. The explicit nature of the solution means that any type of non-linear stress/strain law may be followed without incurring great overhead (as compared to a linear law).

Structures such as tunnel liners, rockbolts, or sheet pilings which interact with the surrounding rock or soil may also be modeled. It is possible, therefore, to examine the stabilizing effects of supported excavations or to examine the effects of soil or rock instability on surface structures.

FLAC Version 2.2 also contains an interface model. In addition to the continuum behavior of the individual zones, slip surfaces or "slide lines" may be used to represent discrete faults or joints within a body upon which the zones may slip or separate.

Version 2.2 of FLAC also offers the ability to model groundwater flow and consolidation. These interactions may be combined with any of the mechanical models.

FLAC is also available with modules for thermal and creep calculations, at an additional cost.

The manual is organized in the following fashion. A simple tutorial is given first, in Section 2, to illustrate operation of the code. Section 3 of the manual provides the theoretical background for the code. In Section 4, a detailed discussion of the input commands is given. This section provides the primary source for information on control of the FLAC program. Section 5 describes the use of FLAC in problem solving. Here, the details of setting up and executing engineering problems are discussed. Section 6 describes the details of using the support-structure interaction logic. In Section 7, example problems which test the various aspects of the code are given, along with comparison to analytical solutions where applicable. Finally, Section 8 discusses common errors encountered in the use of FLAC.

The mathematical details of the constitutive models are given in Appendix A. The use of FLAC for solving consolidation and groundwater flow problems is

described in Appendix F. Graphics or dot matrix printers and the H-P Laserjet are discussed in Appendix G. For FLAC versions with thermal and creep modules, Appendices H and I describe their use.

FLAC Specifications and Installation

FLAC Version 2.2 is available for the following computer systems:

(1) IBM PC- or AT-compatible with

  (a) 8087 or 80287 coprocessor and 640K RAM;

  (b) 80386-based computer with 80287, 80387 or Weitek coprocessor and at least 2 Mb of RAM (Machines with less RAM can run the standard PC or AC version.);

  (c) Definicon Systems Incorporated (DSI) 68020-based family of coprocessor boards;

(2) one (1) floppy disk drive;

(3) one (1) 10 Mb or larger hard disk;

(4) IBM-compatible Color Graphics Adaptor (CGA), Enhanced Graphics Adaptor (EGA), or Video Graphics Array (VGA) monitor;

(5) DOS Version 2.1 or higher; and

(6) Hewlett-Packard or compatible 2- or 6-pen plotter, PostSript laser printer, H-P Laserjet, or dot matrix printer for hard copy of graphics (optional).

With the PC/AT-compatible version, about 2000 zones can be modeled. With a 2 Mb 80386 version, 4300 zones can be used, and the 4 Mb 80386 version can model 15,000 zones.

FLAC Version 2.2 requires approximately 580 Kb to load into RAM. If you have other resident programs, you may have to off-load these programs to run FLAC. Alternatively, a smaller version of FLAC can be provided on request.

FLAC is delivered on one floppy disk which must be restored to the hard disk on your system. The **PKXARC** program supplied on your FLAC disk is used to reassemble the executable file on your hard disk. Copy all the files from the floppy disk onto your hard disk. The following command is then issued:

C> pkxarc flac.arc flac.exe

The computer will now create a "de-compressed" FLAC.EXE file from the FLAC.ARC file.

Several other files are contained on the floppy disk. These are: (1) FLAC.CGA; (2) FLAC.EGA; and (3) several data files. The files FLAC.CGA and FLAC.EGA have FLAC commands which will set your system to the CGA or EGA graphics mode.

As will be discussed later, on start-up, FLAC will look for a file called FLAC.INI in the directory in which FLAC is written. This file contains a series of FLAC commands which set attributes of the system or program which are normally performed every time the code is used. For example, a user who has an EGA system will need to issue the command **SET EGA** upon FLAC start-up to set the attributes of the EGA card. This command may be placed in the FLAC.INI file so that the user need not be bothered with typing it in every time the code is started up. The FLAC.CGA and FLAC.EGA files have the necessary commands to initialize the graphics adaptors. If you have a CGA system (i.e., the standard, low-resolution color monitor), rename the file FLAC.CGA to FLAC.INI:

C> rename flac.cga flac.ini

If you have an EGA system, rename FLAC.EGA to FLAC.INI:

C> rename flac.ega flac.ini

You may add any commands to these files that you wish using any text editor.

FLAC can send plots to a Hewlett-Packard (or compatible) pen plotter, a PostScript-language laser printer, or to a dot matrix printer with graphics ROM. FLAC sends pen plotter commands to a user-selected port. Most pen plotters are connected to serial port COM1. Therefore, the pen plotter output must be directed to COM1 prior to loading FLAC. The serial port must also be initialized to send plots at the desired baud rate. The default is to send plots to COM1 at 9600 baud.

If your plotter is connected to COM2, or if you desire to change the baud rate, simply use the FLAC commands

set output = COM2

and/or

set baud = b

where b = 1200, 2400, 4800 or 9600, to change the default settings. The most convenient way to set these settings is to have the **SET OUTPUT** and **SET BAUD** commands in the file FLAC.INI. Remember, the baud rate dip switch settings on your plotter must be set to coincide with the serial port initialization. Check your plotter manual for instructions on these settings. FLAC assumes U.S. paper size; the plotter should be set accordingly. It is also possible to send output to the parallel port by using the FLAC command

set output = LPT1

As an alternative to pen plotter graphics, bit-mapped graphics may be sent to a dot matrix printer. The DOS graphics program must first be loaded. Before loading FLAC, type

C> graphics

When using FLAC, you may dump a CGA screen graphics image to the dot matrix printer by pressing the shift and print screen keys at the same time. High-resolution dot matrix EGA screen dumps can be made using commercially-available packages.

FLAC can also be used to create a file which can then be run through the **POP** program provided with FLAC to obtain high resolution graphics on many dot matrix printers and on H-P Laserprinters.

Trouble Shooting Plotter Problems

1. The most common error encountered in using pen plotters is a failure to set compatible baud rates for the serial port and pen plotter. Make certain that the dip switch settings on the pen plotter are set properly for the desired baud rate.

2. Make certain that the plotter is connected to the proper serial port specified in the file PLOTTER.BAT.

3. If the non-U.S. paper size is chosen on the pen plotter dip switches, lettering on the plots may be distorted. Make certain that U.S. paper size is selected.

2.0 INSTANT GRATIFICATION — A SIMPLE TUTORIAL ON USING FLAC

This section is provided for the user who wishes to jump in and begin experimenting with FLAC. A simple, fast-running example problem is chosen which demonstrates some advanced aspects of the FLAC code.

2.1 A Trench Constructed in a Cohesionless Soil

A simple problem which can be examined interactively and yet illustrate some of the power of the FLAC code is a trench which is excavated in a cohesionless soil mass.

Execute FLAC by typing:

```
C> flac
```

The computer will load the program and display an initial heading followed by the interactive prompt **flac>**.

To set up the initial finite difference grid, use the **grid** command*:

```
flac> grid 5,5
```

This command will create an initial grid (or mesh, if you prefer) that is 5 zones (or elements) wide by 5 zones high. Now, give the zones a material model and properties. For this example, we use

---

*See reference command list for further details.

the Mohr-Coulomb elasto-plastic model. Type in the following commands:

```
flac> model mohr

flac> prop  bulk=1e8  shear=.3e8  fric=35

flac> prop  dens=1000  coh=1e10
```

Here, we have specified the Mohr-Coulomb model (as you will see later, all of the letters of a command need not be typed in to be "recognized" by the computer). Every zone in the grid could conceivably have a different material model and property. However, by not specifying a range of zones directly behind the **MODEL** command, FLAC assumes that all zones are to be Mohr-Coulomb. The properties are given next --including the bulk modulus (in Pa, note that only b= or simply b followed by the value is all that is required; the full name is given here for clarity), shear modulus, the angle of internal friction, and the cohesion. You see that a very high cohesion value is given. This is only an initial value which is used during the development of gravitational stresses within the body. In effect, we are forcing the body to behave elastically during the initial development of the gravitational stresses.*

*Alternatively, an elastic model could initially be used to set up the virgin stresses, followed by changing the model to Mohr-Coulomb prior to any excavation, applied loads, or other simulations.

This avoids any plastic yield during this initial phase of the model. The reasons for this will become obvious once you gain experience with the explicit simulation procedure.

Now that a grid and model properties have been defined, data pertaining to the simulation can be plotted or printed. Issue the following command:

```
flac> print x  y
```

The x- and y- coordinates will appear in tabular form in the physical positions of the gridpoints. You will note that the table has i (column) and j (row) going from 1 to 6 along the top and left-hand edge of the table. Therefore, each gridpoint and zone has an i (column) and j (row) associated with it. In this example, the gridpoint range is i from 1 to 6 and j from 1 to 6, whereas the zones range from 1 to 5 for i and 1 to 5 for j. If you require greater clarification on this point, see Fig. 4-1 in Chapter 4. To see a plot of the grid, give the following command:*

```
flac> plot grid
```

You will see a plot of the grid on the screen in the low resolution color or EGA mode, depending on your FLAC.INI file. After viewing, press return to get back to the FLAC prompt level. Note that,

*The plotting window will be set automatically unless otherwise specified.

if you do not assign coordinates to the grid (by using the **GEN** or **INITIAL** command), then the x- and y-coordinates are assigned equal to the number of the gridpoint minus 1. For example, in the previous grid plot, the lower left-hand gridpoint is assumed to be the origin and is given the coordinate (0,0). The bottom right-hand corner [gridpoint (6,1)] is given the coordinate (5m,0). The user is completely free to assign any coordinates that he or she chooses by using the **GEN** and **INITIAL** commands. To keep this example simple, we leave the grid at 5mx5m.

Next, the boundary conditions for the problem are set. In this problem, we want to place roller boundaries on the bottom and sides, apply gravitational forces to the zones, and allow the in-situ stresses to develop as they occur in nature. To fix these boundaries (i.e., no displacement or velocity in the specified direction), use the following commands:

```
flac> fix y  j=1
flac> fix x  i=1
flac> fix x  i=6
```

The commands noted above perform the following functions.

1. The bottom boundary gridpoints (j=1) are fixed in the y-direction. When FLAC sees j=1, it automatically assumes that i ranges from 1 to 6 (i.e., the full range). You can perform the same function by specifying j=1, i=1,6.

2. The left-hand boundary gridpoints (i=1) and right-hand boundary gridpoints (i=6) are fixed in the x-direction. Again, FLAC assumes the full range of the j-direction.

Next, we set the gravity by typing in

```
flac> set grav=9.81
```

where 9.81 m/sec^2 is the acceleration due to gravity. Here, gravity is taken as positive downward and negative upward. (If gravity is set negative, objects will float!)

If you wish to see a history of the displacement of a gridpoint on the model to observe equilibrium or collapse with time, type

```
flac> his nstep=5
flac> his ydis  i=2  j=6
```

Here, we choose to monitor the y-displacement every 5 timesteps for a point at the top of the ground surface.

Now, we are ready to bring the initial model to equilibrium. Because FLAC is an explicit dynamic code, we step the model through time*, allowing the kinetic energy of the mesh to damp out (thus providing the static solution we are after). To

*Calculational time—not real time.

allow gravity to develop within the body, we time-step the simulation to equilibrium. Here, the **SOLVE** command is used to automatically detect equilibrium*:

```
flac> solve
```

FLAC will "think" for a period of time (a few minutes). At each step in time, the step number and maximum out-of-balance force will appear on the screen. When completed, FLAC will return with a message that the solution limit has been reached. The FLAC prompt will then reappear.

Now, we can see what has occurred within the model. Examine the y-displacement history requested earlier:

```
flac> plot his 1
```

A screen plot will be shown in monochrome or EGA mode which indicates that the model came to equilibrium within roughly 108 timesteps.

*See Chapter 4 for setting the limits on the **SOLVE** command.

Let's examine the gravitational stresses developed in the body. The window is automatically defined but, if you wish to enlarge or shrink the plot, you can reset it. Now, give the plot a title* by typing

```
flac> title
title> a simple trench excavation example:  the
          initial gravity stresses
```

Type the following (if you are in monochrome mode, leave off the color keywords):

```
flac> set pal=0
```

(to set the palette choice), then

```
flac> plot syy  yel  bou  gre
```

You will see a plot (Fig. 2-1) of the σ_{yy} stresses in yellow-brown and the boundary in green (all is green on the monochrome screen). Similarly, the σ_{xx} stresses can be plotted by typing

```
flac> plot sxx  yel  bou  gre
```

You will note that the gravitational stresses increase linearly with depth. These can be identified by typing

```
flac> print sxx  syy
```

---

*The title and legend appear on hard-copy pen plots but not on screen plots.

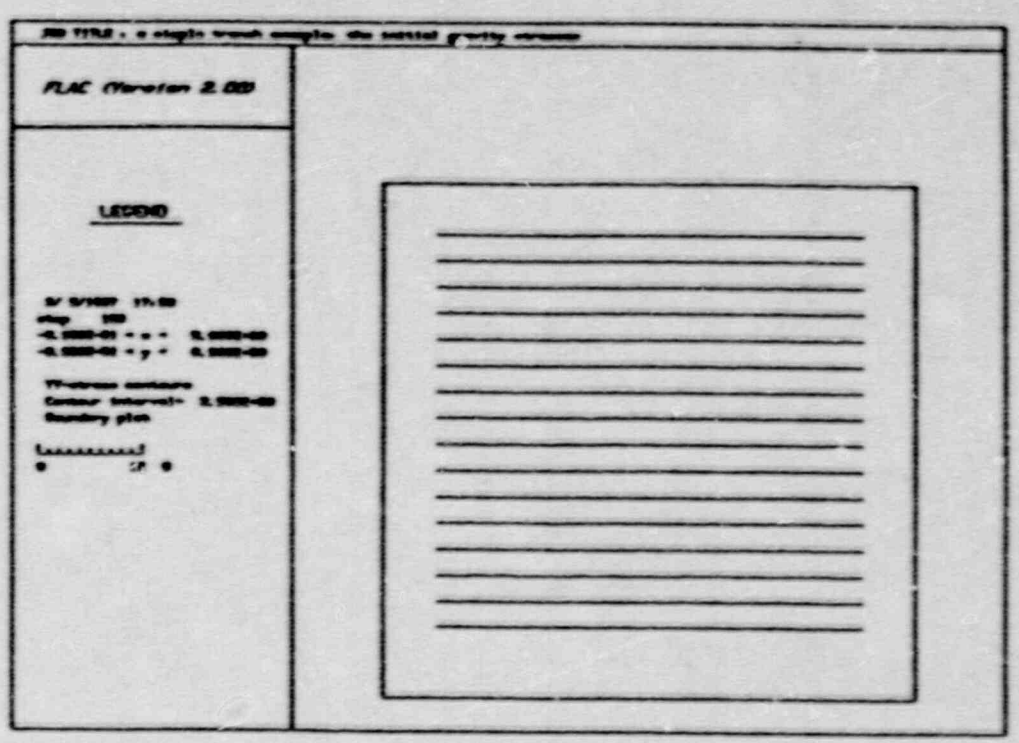


Fig. 2-1 The Gravitational Stresses Induced in the Soil After 108 Timesteps

It is wise to save this initial state so that you can restart it at any time for performing parameter studies. To save this, type

```
flac> save trench.sav
```

A save file will be created on the default drive. A FLAC prompt will then return. Now we can excavate a trench in the soil. Type

```
flac> prop coh=0
```

With a zero cohesion and vertical, unsupported trench walls, collapse will certainly occur. Because we want to examine this process accurately, the large strain logic must be set in the code. This is done by typing

```
flac> set large
```

Finally, for plotting purposes, we wish to see only the change in displacements from the trench excavation and not the previous gravitational setting—so we can zero out the x and y components:*

```
flac> init xdis=0  ydis=0
```

To excavate the trench, enter

```
flac> model null  i=3  j=3,5
```

*This will not affect the calculations since the model does not require displacements in the calculation sequence. They are kept only as a convenience to the user.

Because we purposely set the cohesion low enough to result in failure, we do not want to use the **SOLVE** command with a limit for out-of-balance forces* (which checks for equilibrium), as our simulation will never converge to the equilibrium state. Instead, we can step through the simulation process one timestep at a time and plot and print the results of the collapse as it occurs. This is the real power of the explicit method—the model is not required to converge to equilibrium at each calculation cycle because we never have to solve a set of linear algebraic equations, as is the case in the implicit codes with which most engineers are familiar. To do this, we use the **STEP** command:+

```
flac> step 100
```

FLAC will now step through 100 timesteps. When it is finished, the prompt will reappear. Now, examine the results thus far by printing and plotting some variables—e.g.,

```
flac> pr state
```

or

```
flac> plot state
```

---

*See Chapter 4, **SOLVE** command, for more details.

+The **SOLVE** command can also be used with limits on the number of timesteps.

The present yield state of each zone will be printed in tabular form. The number 1.0 means that the material of this zone is actively yielding. A value of 0.0 means that the zone has always been elastic. As you can see, the zones adjacent to the trench are actively yielding.*

Now, try plotting some parameters:

```
flac> plot grid
```

You will notice some grid distortion beginning at the trench, but the resolution of the screen will be a bit poor for accurate plot reproduction. If you have a Hewlett-Packard pen plotter, try sending a plot—you will see improved resolution. (See the **PLOT** command in Chapter 4 for details.)

Next, try some plot overlays to define better the extent of the failure area. (If you wish, first retitle the plot using the **TITLE** command.)

```
flac> plot xv z yell dis red max=1e-2 bo green
```

You will see a plot of the x-velocity contours (in yellow, zero contours removed) overlain by the displacement contours (in red, scaled to a maximum vector length of 1×10^{-2} m) and the boundary (in green). This is shown in Fig. 2-2. The velocity

*Note that we have made our boundaries on this problem small in order to speed operation; thus, some boundary interference occurs.

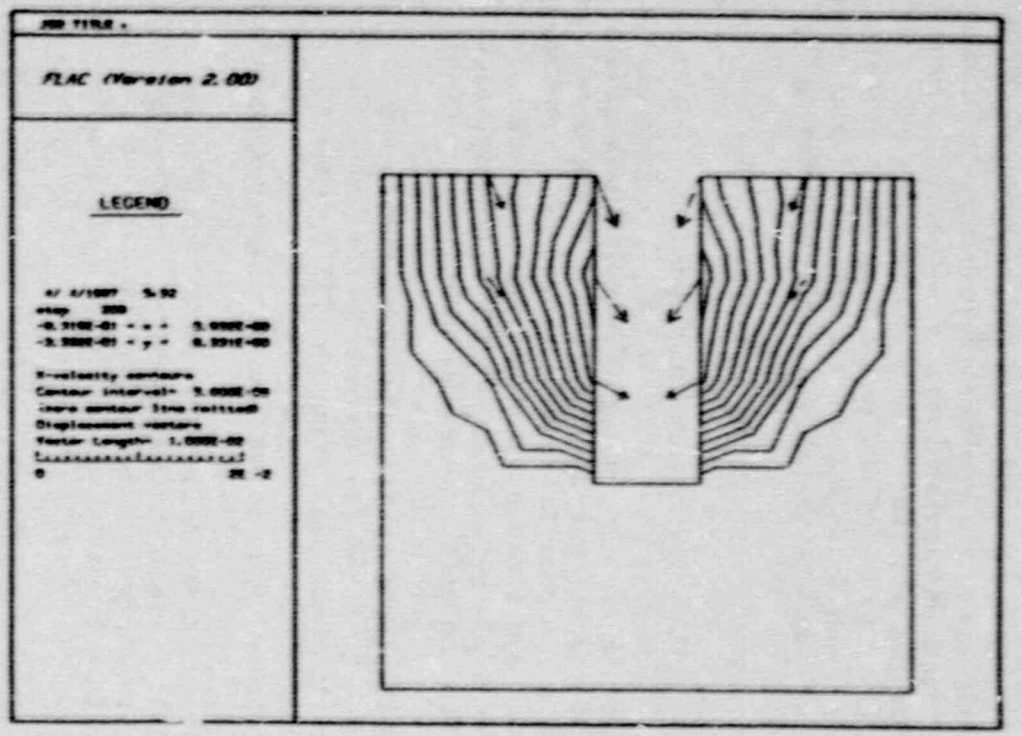


Fig. 2-2 A Plot of the Displacement Vectors and x-Velocity at Timestep 208

contours are given here to help visualize those areas of active yield, since this material is flowing.

The collapse process can be examined as it occurs by timestepping 100 steps at a time. Note that you can stop the FLAC calculation at any time with any keystroke. FLAC will return control to you after it has completed the current step. We encourage you to step ahead in this fashion, creating plots and experimenting with the **MAX**, **INIT** and **COLOR** commands at each stage. Try plotting the stresses, velocities and displacements to produce meaningful plots. In this example, we will jump ahead to a convenient spot in the collapse process:

```
flac> step 400
```

Again, try

```
flac> plot grid
```

You will see a drastically different picture at this stage as the trench collapses (Fig. 2-3). By typing

```
flac> pr state
```

you will note that the zones are still at the yield point. Examine the σ_{yy} state and displacements by requesting

```
flac> plot syy zero disp mar=0.2 red bou gree
```

You will note distortion of the stress contours due to the excavation and an increase in magnitude (by approximately 20 times) of the displacement vectors.

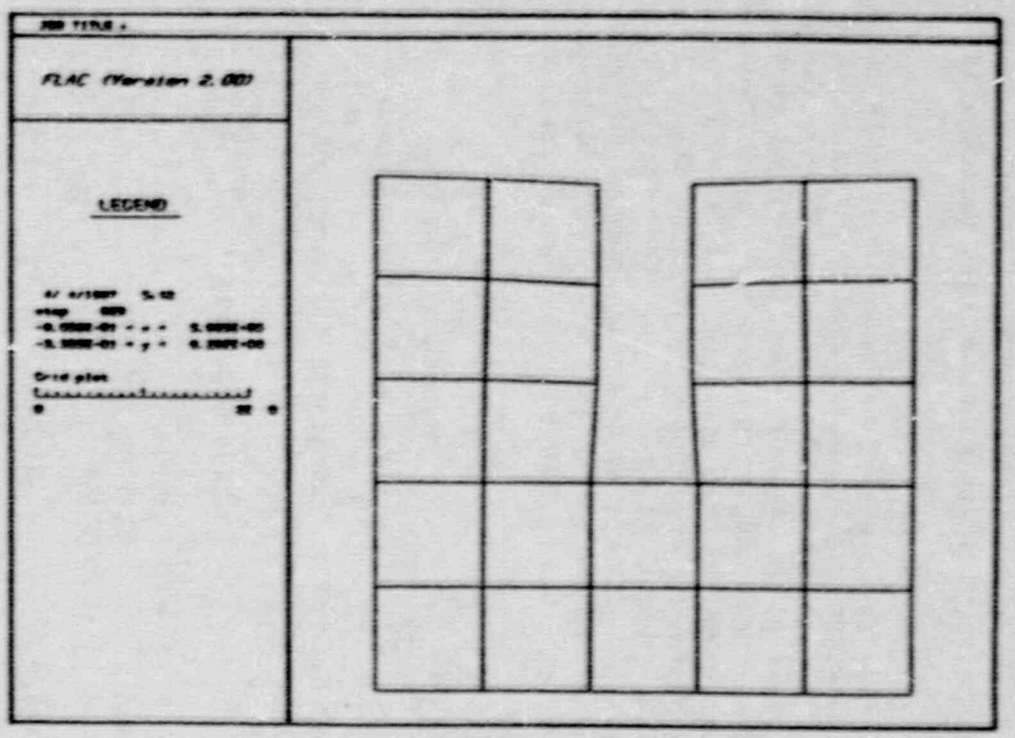


Fig. 2-3 Deformed Mesh After 608 Timesteps

From this point, you may wish to play with the various features of FLAC in an attempt to stabilize the excavation. Try restarting the previous file you created by entering

flac> rest trench.sav

Excavate the trench as before but try using the structural element logic described in Section 6 to model bracing or tie-back anchors.

You will see that FLAC is virtually bullet-proof —an error-trapping function recognizes most commonly-occurring errors.

This ends the initial tutorial. Section 5 contains additional details on problem solving with FLAC.

3.0 BACKGROUND — THE EXPLICIT FINITE DIFFERENCE METHOD

3.1 Introduction

FLAC is a general geomechanics numerical model which uses the explicit finite difference method to solve the basic equations of motion. The finite difference method, in general, involves division of the body to be modeled into a number of two-dimensional zones (elements) which are interconnected at their gridpoints (nodes). At each gridpoint, the form of the equations of motion are solved in a timestepping fashion. It is possible, therefore, to see the behavior of a system as it evolves with time.

Several authors (among them, Wilkins, 1963) have shown that the equations resulting from the finite difference and finite element methods are identical for particular examples. The finite difference method is as flexible in use as is the finite element method in that irregular meshes may be used and varying material models or properties and various boundary conditions may be specified. Particular applications of finite differences to geomechanics problems are discussed in Cundall (1976).

3.2 Explicit Formulation/Calculation Cycle

There are, in general, two methods employed by numerical codes in the solution of the equations of motion for a particular problem. The implicit approach (used in many numerical methods to solve static problems) solves for unknown values at all gridpoints at one time. In other words, a system

of equations is set up which relates unknown to known quantities such as the relationship of nodal forces to displacements via the global stiffness matrix in a finite element model. This system of equations must be stored and solved, resulting in large computer memory requirements.

The explicit method makes use of the idea that, for small timesteps, a disturbance at a given gridpoint is experienced only by its immediate neighbors. As an example, imagine that the temperature of a gridpoint is raised to some value at time t=0. For a short period of time, only the neighboring gridpoints "realize" that the temperature has been increased. As time goes on, the effect will spread through the grid, resulting in higher temperatures at surrounding grid points.

The timestep must be chosen carefully to avoid numerical instability in the solution—e.g., the timestep must be less than the time of propagation of the phenomenon between two adjacent gridpoints. In the mechanical version of FLAC, this timestep is governed by the speed of sound through the body; in the thermal version of the code, the thermal diffusivity and convective heat transfer coefficients govern the thermal timestep. FLAC automatically determines a timestep which ensures numerical stability.

At present, FLAC is used to solve static problems by properly damping the dynamic solution. In this case, "timestep" refers not to a physical timestep but a problem timestep, where velocities are measured in length per timestep. This solution procedure results in many advantages for modeling of geological materials on the personal computer.

Because matrices are never formed, the memory requirements of the code are minimal and the computational effort per timestep is small. A very important aspect of the explicit method for geological materials is the ease with which non-linear constitutive laws are handled. No iterations (which may cause significant errors in the solution) are required for the material to follow non-linear stress-strain laws; the stress change corresponding to a given strain change may simply be enforced at a given zone as occurs in reality. In this manner, non-linear laws are followed in the correct physical fashion and are not dependent on the path sensitivity of iterative methods. Also, because matrices are not formed, phenomena such as large displacements can be handled with little additional computational effort. On the negative side, however, explicit codes are, in general, slower to run than their implicit counterparts for static, elastic problems.

What does all this mean for modeling geotechnical materials? For soils and rocks, the material very often undergoes failure or yield (i.e., the material is non-elastic) and may exhibit large displacements. Thus, a body may experience collapse in an isolated region (e.g., a soil slope) yet be stable in the remainder of the body. For such a problem, an implicit code, in which all gridpoints are interdependent would become numerically unstable during matrix inversion. However, the explicit code suffers no time or numerical stability losses. A more subtle advantage is that the user can examine the development of yield or material collapse as it develops instead of viewing the end equilibrium state only.

The explicit calculation cycle employed by FLAC is illustrated in Fig. 3-1. For each timestep, the equations of motion are solved for each gridpoint in the mesh. The forces present at each gridpoint are not in balance for a non-equilibrium state. This gives rise to accelerations of the gridpoint based on the components of the out-of-balance force and the zone mass lumped at the gridpoint. The accelerations are integrated to yield the gridpoint velocities which are, in turn, used to determine the strain change.

The strains are used in the constitutive law to determine the corresponding stress increment for the zone. Once the stress increments have been determined, the forces which they produce on the surrounding gridpoints are summed to determine the resulting out-of-balance force.

This calculation cycle is repeated every timestep. The user may examine the present state of the problem at any step in this process. As stated earlier, the dynamic response of the system is numerically damped within the FLAC code to provide a static solution. The method of damping employed in the code is discussed later. The damping results in the decay of the out-of-balance forces to zero as the problem approaches static equilibrium with increasing timestep. The number of timesteps required for equilibrium depends on many factors,

---

*The velocities are integrated to yield the displacements which are kept as a convenience to the user.

At each time step:

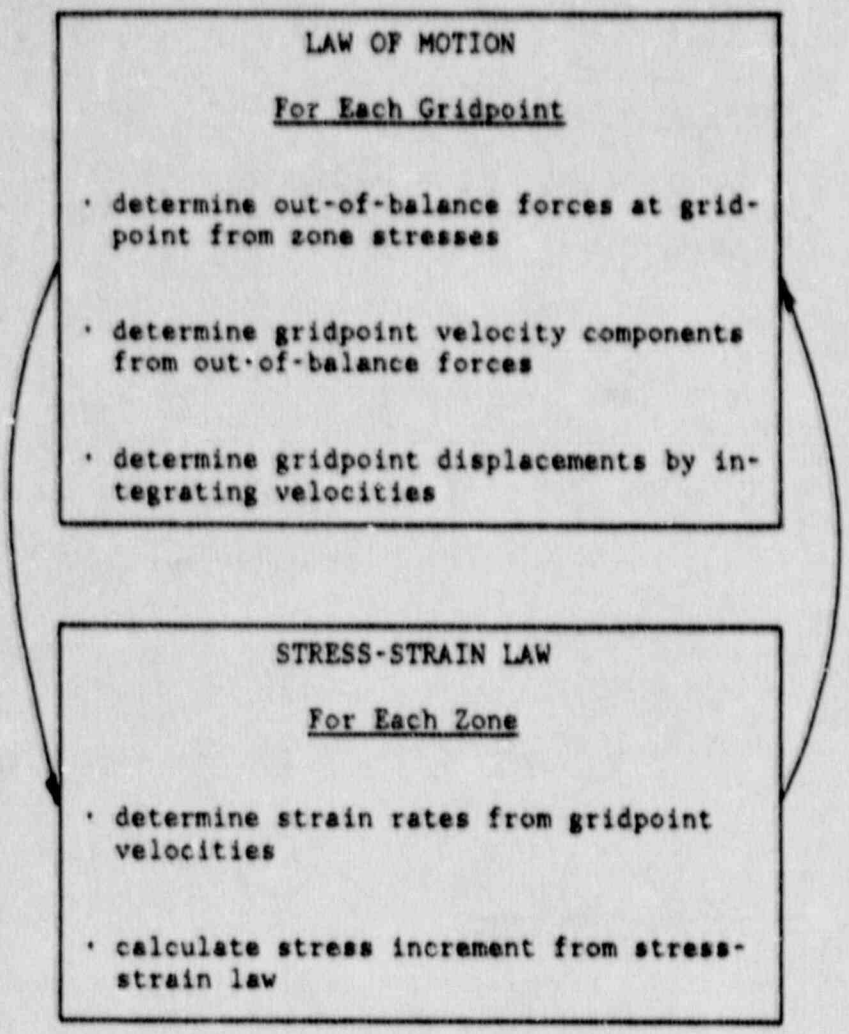


Fig. 3-1 Overview of Code Operations

including the solution accuracy desired by the user, the extent of yield in the material, and the size of the problem.

3.3 Finite Difference Scheme*

3.3.1 Representation of Derivatives — Wilkins (1963) presents a differencing scheme based on the integral definition of a partial derivative:

$$\frac{\partial F}{\partial x_i} = \lim_{A \to 0} \left[\frac{1}{A} \int_s F\, n_i\, ds \right] \qquad (3\text{-}1)$$

where F = a scalar, vector or tensor;

x_i = components of position vector;

A = area of integration;

ds = incremental arc length; and

n_i = components of the unit normal to ds

*The discussion given here follows lecture notes by P. A. Cundall presented at the University of Minnesota (1983).

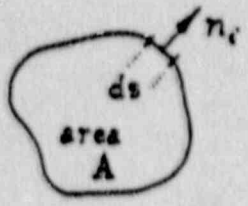


Fig. 3-2 Integration Path

The surface integral of Eq. (3-1) is continuous, but an equivalent (although approximate) expression may be written for integration done over a finite polygon:

$$\frac{\partial F}{\partial x_i} = \frac{1}{A} \sum_{n=1}^{N} \overset{n}{F} \varepsilon_{ik} \overset{n}{\Delta x_k} \qquad (3\text{-}2)$$

where N = number of sides;

$\overset{n}{F}$ = average value of F on side n;

$\overset{n}{\Delta x_i}$ = components of vector length of side n; and

ε_{ik} = permutation tensor in two dimensions,

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

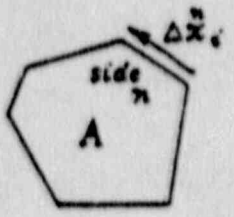


Fig 3-3 Discretization of Boundary for Numerical Integration

Expression (3.2) is used to derive all spatial difference equations in FLAC. It should be noted that this formulation imposes no restriction on the shape or number of sides, unlike many finite difference formulations based on rectangular grids.

3.3.2 Equations of Motion — The equations of motion are given by the familiar expression

$$\rho \left[\frac{\partial \dot{u}_i}{\partial t} \right] = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i \qquad (3\text{-}3)$$

where ρ = density;

$\dot{u}_i$ = velocity;

t = time;

σ_{ij} = stress tensor; and

g_i = body force components.

As a simple example, examine the motion of a mass subjected to a time-varying force, F (Fig. 3-4):

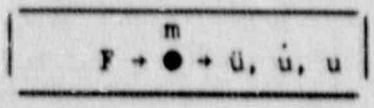


Fig 3-4 Mass Subject to a Time-Varying Force, F

$$\frac{\partial u}{\partial t} = \frac{F}{m} \quad \text{(i.e., } F = ma\text{)} \qquad (3\text{-}4)$$

A central difference* scheme may be used to solve Eq. (3-4) by storing velocities at the half timestep. The acceleration [left-hand side of Eq. (3-4)] can be written as

$$\frac{\partial \dot{u}}{\partial t} = \frac{\dot{u}^{(t + \Delta t/2)} - \dot{u}^{(t - \Delta t/2)}}{\Delta t} \qquad (3\text{-}5)$$

Substitution into Eq. (3-4) yields

$$\dot{u}^{(t + \Delta t/2)} = \dot{u}^{(t - \Delta t/2)} + \left[F^{(t)}/m \right] \Delta t \qquad (3\text{-}6)$$

which is the velocity of the gridpoint at the half timestep. Now, the displacement can be determined from the velocity by one additional integration:

$$u(t + \Delta t) = u(t) + \dot{u}^{(t + \Delta t/2)} \Delta t \qquad (3\text{-}7)$$

The force can be determined during the same time increment, resulting in the calculation sequence illustrated in Fig. 3-5.

---

* A Taylor's series can be used to show that a central difference scheme is second-order accurate—i.e., error terms in first order Δt will vanish. See, for example, Jaeger and Starfield (1974).

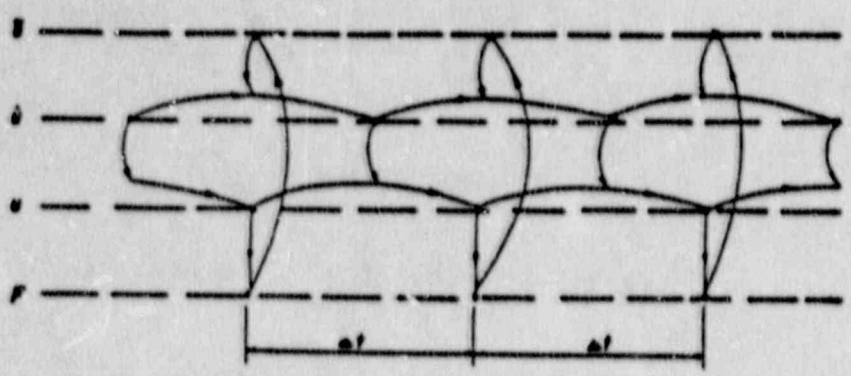


Figure 3-5 Interlaced Nature of Calculation Cycle Used in FLAC

3.4 Velocity/Strain Increment Equations

FLAC determines an initial strain increment at each timestep from the velocities at each grid-point. The strain increment is used in the chosen constitutive laws to determine the corresponding stress increment. In incremental form, the strain tensor is given by

$$\Delta e_{ij} = \frac{1}{2}\left[\frac{\partial \dot{u}_i}{\partial x_j} + \frac{\partial \dot{u}_j}{\partial x_i}\right]\Delta t \qquad (3\text{-}8)$$

where Δe_{ij} = strain increment tensor
$i,j = 1,2$;

$\dot{u}_i$ = i component of the velocity
$i = 1,2$;

x_i = i component of the coordinate
$i = 1,2$; and

Δt = timestep.

After the application of the equation of motion [Eq. (3-3)], a set of x- and y-velocities exist at each corner gridpoint of a typical quadrilateral element. FLAC subdivides this quadrilateral into two pairs of superimposed constant-strain triangular zones termed a, b, c and d (Fig. 3-6):

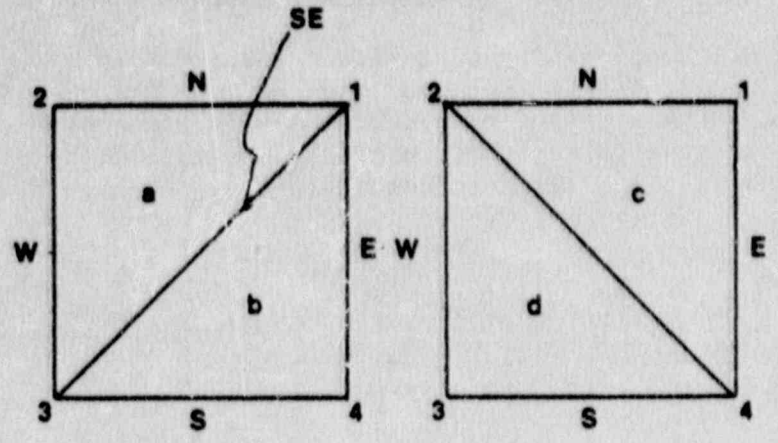


Fig. 3-6 Constant-Strain Triangular Zones in FLAC

Recall [Eq. (3-8)] that the velocity gradients are required when calculating the strain increment tensor. Based on Eq. (3-2), the partial derivatives may be replaced by their difference form:

$$\frac{\partial \dot{u}_i}{\partial x_j} = \frac{1}{A} \sum_{\text{edges}} \dot{u}_i \, \epsilon_{jk} \, \Delta x_k \qquad (3\text{-}9)$$

This summation is taken over the edges of the zone, but velocities are known only at the corners. If the average velocity components are taken along each edge, the term inside this summation is identical to that found by exact integration from Eq. (3-1), assuming a linear variation of velocity along the edge. FLAC determines the velocity gradients initially for triangles a and b and then for triangles c and d. For a, the expansion of Eq. (3-9) gives

$$\frac{\partial \dot{u}_i}{\partial x_j} = \frac{1}{2} \frac{1}{A} \left[\begin{array}{l} (\dot{u}_i^{(1)} + \dot{u}_i^{(2)}) \, \epsilon_{jk} \, \Delta x_k^{(N)} \\ + (\dot{u}_i^{(2)} + \dot{u}_i^{(3)}) \, \epsilon_{jk} \, \Delta x_k^{(W)} \\ + (\dot{u}_i^{(1)} + \dot{u}_i^{(3)}) \, \epsilon_{jk} \, \Delta x_k^{(SE)} \end{array} \right] \qquad (3\text{-}10)$$

To illustrate the above expansion for one component of the derivative, examine the xx-strain rate

$$\frac{\partial \dot{u}_1}{\partial x_1} = \frac{1}{2}\,\frac{1}{A}\left[\begin{array}{l}(\dot{u}_1^{(1)} + \dot{u}_1^{(2)})\,(x_2^{(2)} - x_2^{(1)}) \\ + (\dot{u}_1^{(2)} + \dot{u}_1^{(3)})\,(x_2^{(3)} - x_2^{(2)}) \\ + (\dot{u}_1^{(1)} + \dot{u}_1^{(3)})\,(x_2^{(1)} - x_2^{(3)})\end{array}\right] \qquad (3\text{-}11)$$

Expanding and cancelling terms gives

$$\frac{\partial \dot{u}_1}{\partial x_1} = \frac{1}{2}\,\frac{1}{A}\left[\begin{array}{l}\dot{u}_1^{(1)}\,(x_2^{(2)} - x_2^{(3)}) \\ + \dot{u}_1^{(2)}\,(x_2^{(3)} - x_2^{(1)}) \\ + \dot{u}_1^{(3)}\,(x_2^{(1)} - x_2^{(2)})\end{array}\right] \qquad (3\text{-}12)$$

All other velocity gradient components are determined in the same manner and used to determine the three components of the strain increment tensor.

The problem of "hourglassing" deformation modes sometimes experienced in the use of finite differences is eliminated here because triangular zones are used in the calculations rather than quadrilaterals.

In order to prevent "lock-up" during plastic flow, the mixed-discretization scheme of Marti and Cundall (1982) is used between triangles a and b and between c and d. The forces provided by the two sets of triangles are averaged at the gridpoints.

3.5 Stress-Strain Law

Once the strain tensor has been determined in each triangular element, the corresponding stress must be calculated from the chosen constitutive law for the zone. Prior to entering the constitutive law, two corrections to the existing stresses are required. First, if large strain logic is in use, the stresses must be corrected for rotations induced by displacement of the gridpoints. These are discussed in Section 3.6.

Second, if effective stresses are used the pore pressure must be subtracted from the direct stress components.

The constitutive relation between strain and stress is given in incremental form:

$$\Delta\sigma_{ij} = f\,(\Delta e_{ij},\ \sigma_{ij},\ \ldots) \qquad (3\text{-}13)$$

where $\Delta\sigma_{ij}$ = stress increment tensor i,j

= 1,2, and

$f(\ )$ = a constitutive equation involving incremental strains, existing total stress state, material constants, etc.

The details of the constitutive laws and the calculation of stress components from the strains are given in Appendix A.

Finally, the average stress components for a quadrilateral are determined by a weighted average of the triangular components. FLAC allows for printing of either the triangular or averaged stress components.

3.6 Determination of Out-of-Balance Force at a Gridpoint

Once the stress tensor within the quadrilateral zone has been determined, it can be used to calculate the out-of-balance forces at the gridpoints. Recalling the equation of motion,

$$\rho \left[\frac{\partial \dot{u}_i}{\partial t} \right] = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i$$

note that the right-hand side may be replaced by Eq. (3-2) and divided through by ρ. The result is

$$\rho \left[\frac{\partial \dot{u}_i}{\partial t} \right] = \frac{1}{A} \Sigma \sigma_{ij} e_{jk} \Delta x_k + \rho g_i \qquad (3\text{-}14)$$

or

$$\frac{\partial \dot{u}_i}{\partial t} = \frac{F_i}{m} + g_i$$

where $F_i = \Sigma\, \sigma_{ij}\, \epsilon_{jk}\, \Delta x_k$; and $m = \rho A$.

All vector quantities are taken to occur at gridpoints and, therefore, the mass, m, in the above equation is taken to be a lumped mass with contribution from the surrounding zones (Fig. 3-7). The summation term of Eq. (3-14) is taken to occur over the closed contour illustrated in this figure.

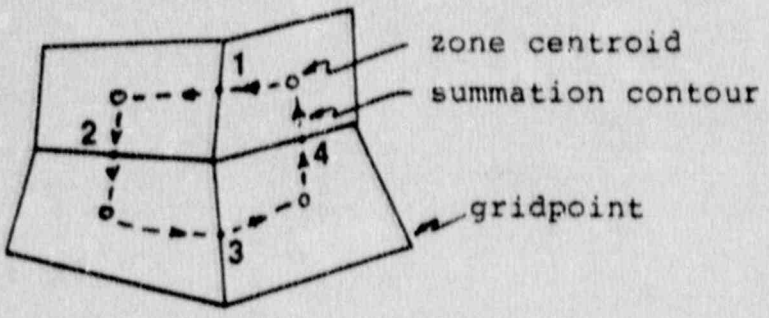


Fig. 3-7 Closed Contour for Summation

If the closed contour is taken over the path shown in this figure and the lumped mass is assumed to be equal to the mass enclosed by the path, then

(1) the total area of the grid is exactly covered by all such paths around gridpoints, and

(2) the lumped mass fulfills certain conditions—namely, conservation of center of gravity.

Although the path of a contour within a zone is important for the mass lumping, it is unimportant for the summation term in Eq. (3-14). Because zone stress is constant over a zone, the summation gives the same result for any path, provided that the endpoints are fixed (at the centers of grid lines). Hence, the gridpoint force, as illustrated in Fig. 3-7, becomes

$$F_i = {}^{a}\sigma_{ij}\,\epsilon_{jk}\,(x_k^{(1)} - x_k^{(4)}) + {}^{b}\sigma_{ij}\,\epsilon_{jk}\,(x_k^{(2)} - x_k^{(1)}) + {}^{c}\sigma_{ij}\,\epsilon_{jk}\,(x_k^{(3)} - x_k^{(2)}) + {}^{d}\sigma_{ij}\,\epsilon_{jk}\,(x_k^{(4)} - x_k^{(3)}) \qquad (3\text{-}15)$$

This force, and all the stresses and coordinates, are known at times t, t + Δt, etc. The gridpoint velocity at the next half timestep is

$$\dot{u}_i^{(t + \Delta t/2)} = \dot{u}_i^{(t - \Delta t/2)} + \frac{F_i^{(t)}}{m} \Delta t + g_i \qquad (3\text{-}16)$$

It should be noted that if any of the zones in Fig. 3-7 are missing (e.g., at a boundary or if the zone is null), the term corresponding to the missing zones is simply omitted from Eq. (3-15).

If the large-strain mode of FLAC is selected, the new velocity in Eq. (3-16) is used to update the grid coordinates:

$$x_i^{(t + \Delta t)} = x_i^{(t)} + \dot{u}_i^{(t + \Delta t/2)} \Delta t \qquad (3\text{-}17)$$

A displacement vector is also updated in a similar fashion for the convenience of the user, although displacements are not used by FLAC.

3.7 Stress Rotation Correction Terms

When using the large-strain logic in the code, the stresses must be corrected for rotation of the body. The regular transformation equation for stress is

$$\bar{\sigma}_{ij} = \sigma_{\alpha\beta} J_{i\alpha} J_{j\beta} \qquad (3\text{-}18)$$

where $J_{ij} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ for 2-D, and

$\bar{\sigma}_{ij}$ = stress tensor corrected for large strain rotations

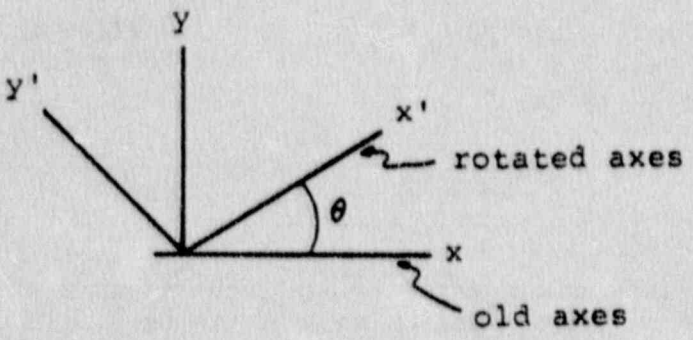


Fig. 3-8 Rotation of Stresses by an Angle θ

For small angles $\Delta\theta$,

$$J_{ij} = \begin{bmatrix} 1 & \Delta\theta \\ -\Delta\theta & 1 \end{bmatrix},$$

since $\cos\Delta\theta \approx 1$ and $\sin\Delta\theta \approx \Delta\theta$. Hence,

$$\bar{\sigma}_{11} = \sigma_{11}J_{11}J_{11} + \sigma_{21}J_{12}J_{11} + \sigma_{12}J_{11}J_{12} + \sigma_{22}J_{12}J_{12}$$

$$= \sigma_{11} + \sigma_{21}\Delta\theta + \sigma_{12}\Delta\theta + \sigma_{22}\Delta\theta^2 \qquad (3\text{-}19)$$

$\Delta\sigma_{11} = 2\sigma_{12}\Delta\theta$, since $\sigma_{21} = \sigma_{12}$, and

$\Delta\theta^2 = 0$ for small $\Delta\theta$.

This is a change in stress, referenced to the new axis at $\Delta\theta$ from the old axis—but we want the new stress referenced to the old, original, axis. Hence, the correction terms for σ_{11} is

$$\Delta\sigma_{11} = -2\sigma_{12}\Delta\theta \qquad (3\text{-}20)$$

Similarly,

$$\bar{\sigma}_{11} = \sigma_{11}J_{21}J_{21} + \sigma_{21}J_{22}J_{21} + \sigma_{12}J_{21}J_{22} + \sigma_{22}J_{22}J_{22}$$

$$= \sigma_{11}\Delta\theta^2 - \sigma_{21}\Delta\theta - \sigma_{12}\Delta\theta + \sigma_{22}$$

$$= \sigma_{22} - 2\sigma_{12}\Delta\theta + \sigma_{11}\Delta\theta^2 \qquad (3\text{-}21)$$

Reversing sign and ignoring the term in $\Delta\theta^2$,

$$\Delta\sigma_{22} = 2\sigma_{12}\Delta\theta \tag{3-22}$$

Also,

$$\begin{aligned}\bar{\sigma}_{12} &= \sigma_{11}J_{11}J_{21} + \sigma_{21}J_{12}J_{21} + \sigma_{12}J_{11}J_{22} \\ &\quad + \sigma_{22}J_{12}J_{22} \\ &= -\sigma_{11}\Delta\theta - \sigma_{21}\Delta\theta^2 + \sigma_{12} + \sigma_{22}\Delta\theta\end{aligned} \tag{3-23}$$

$$\Delta\sigma_{12} = (\sigma_{11} - \sigma_{22})\Delta\theta \tag{3-24}$$

These correction terms are added to the stresses when large strain logic is used.

References

Cundall, P. A. "Explicit Finite Difference Methods in Geomechanics," Numerical Methods in Engineering (Proceedings of the EF Conference on Numerical Methods in Geomechanics, Blacksburg, VA, 1976), Vol. 1, pp. 132-150.

Cundall, P. A. Lecture Notes, University of Minnesota, 1983.

Jaeger, J. C., and A. M. Starfield. An Introduction to Applied Mathematics (2nd Ed.). Oxford: Clarendon Press, 1974.

Marti, J., and P. A. Cundall. "Mixed Discretization Procedure for Accurate Solution of Plasticity Problems," Int. J. Num. Methods in Eng., 6, 129-139 (1982).

Wilkins, M. "Calculation of Elastic-Plastic Flow," Lawrence Radiation Laboratory Report UCRL 7322, 1963.

4.0 INPUT INSTRUCTIONS

4.1 Definitions

A few definitions are reviewed here prior to a detailed discussion of the input commands. Greater detail on terminology and problem set-up are given in Section 5.0.

Zone	an element of the finite difference mesh, analogous to a finite element.
Gridpoint	a corner node of the finite difference zone. Each quadrilateral zone is defined by its four corner gridpoints. Adjacent zones are connected at their gridpoints.
Grid	an assemblage of finite difference zones, analogous to a finite element mesh.

In normal operation, the finite difference mesh origin is in the lower left-hand corner of the grid. The i(x) and j(y) axes begin at this point (Fig. 4-1). Each zone and gridpoint thus have an i (column) and j (row) number associated with them.

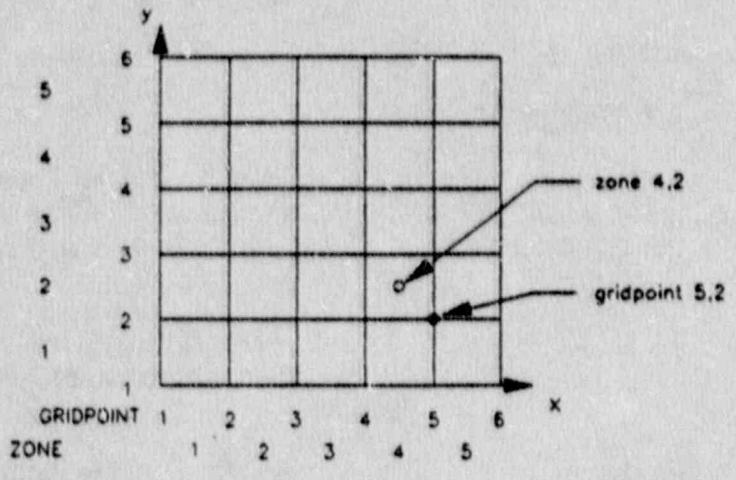


Fig. 4-1 Finite Difference Grid Nomenclature (zones and gridpoints are given i,j coordinates)

Because FLAC is an explicit code, the basic equations of motion are solved at successive steps in time. The code is dynamic, which means that each gridpoint has a velocity component which must be damped to provide the static solution. This damping is done automatically in FLAC as the code is timestepped to equilibrium. The equilibrium solution is obtained when the velocities have reached some sufficiently small magnitude. The means by which equilibrium is determined is discussed in Appendix B.

Setting Your Own Default Conditions

If you wish FLAC to assume certain parameters or modes whenever you start the program, prepare a file called FLAC.INI. This file may contain any valid FLAC command(s). FLAC will read the file automatically on start-up and process the commands. For example, FLAC.INI might contain the following:

```
set force=5e2, EGA, plot=post, output=flac.ps

set water density=1000  aspect=1.05
```

If the file FLAC.INI does not exist, FLAC simply continues without error. Note that some commands in a FLAC.INI file may result in an error message. For example, if you attempt to give properties to a grid prior to defining the grid, the normal error message will arise.

4.2 Input Commands

The input commands to FLAC differ from most computer programs written for numerical modeling; commands are specifically designed for simplicity and ease of use by the program operator. FLAC may be operated in "interactive" mode (i.e., commands entered via the keyboard) or "file-driven" mode (i.e., data stored on a data file and read in from diskette or hard disk). In either case, the commands for running a problem are identical, and the particular method of data input depends on user preference.

All input commands are word-oriented and consist of a primary command word followed by keywords and numerical input, as required. The commands, given on the following pages, are typed literally on the input line. You will note that only the first few letters are capitalized. The program requires only these letters to be typed for the command to be "recognized". Many of the keywords are followed by a series of numbers which provide the numeric input required by the keyword. Words that begin with a lower-case letter stand for numbers. Integers are expected when the word begins with i, j, m or n; otherwise, a real (or decimal) word is expected. The decimal point can be omitted from a real number but must not appear in an integer.

These values may be separated by any number of spaces or by any of the following delimiters:

( ) , / =

You will see additional notations with some of the input parameters. These are:

< >	denotes optional parameter(s) (the brackets are not to be typed);
...	indicates that an arbitrary number of such parameters may be given.

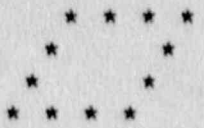
Anything that follows a "*" or ";" in the input line is taken to be a comment and is ignored. It is useful to make such comments in the input file if running in batch (i.e., file-driven) mode since the comments are reproduced on the output.

NOTE: The input command to FLAC may contain a full 80 characters. In the examples given in the following section, the page width restricts the length of the input command. In some cases in the manual it is necessary to continue the input command on the next line. In such cases, the continuation line is indented. These commands are typed on a single line in the FLAC program.

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

positive for:

motion upwards and to the right
tensile stress
extensional strain
shear strain as follows:



The sign for a force boundary condition is determined by the direction of the force vector (i.e., + when pointing in the positive axis direction). Although we recommend SI units, any consistent set of engineering units for properties may be used. A review of systems of units is given in Table 4-1. No conversions are performed in the program except for friction and dilation angles, which are entered in degrees.

The following examples illustrate the use of the input commands. To fix the x-direction displacement (or velocity) for all gridpoints in the range of i=1 to 10 and j=1, the following commands produce the same result:

```
fix x  i=1,10  j=1,1
```

or

```
fix x  j=1 i= 1 10
```

or

```
f  x  i=1,10  j=1
```

If either the i range or the j range is omitted, the whole range of the grid is assumed.

Another example illustrates the use of optional parameters in an input line. To create a plot of the displacement vectors scaled to a maximum length of 0.001, the following command is given:

plot disp/max=.001

When the program "recognizes" the word "displacement", it looks for additional descriptive or "switch" words. In this case, the word "max" sets the maximum value of the displacement vectors for plotting. Note, also, that, in several cases, many keywords can be given on the same input line.

For example,

plot disp/max=.001 sxx/int=10E6/red

will produce a plot of displacement vectors as above, superimposed with the σ_{xx} stress contours with contour intervals of 10^7, drawn in red. Note that the / and = in the above examples are used as delimiters strictly for the convenience of the user and may be eliminated if desired.

Table 4-1

SYSTEMS OF UNITS

	METRIC				BRITISH	
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	lb_f	lb_f
Stress	Pa	kPa	MPa	bar	lb_f/ft^2	psi
Gravity	m/sec^2	m/sec^2	m/sec^2	cm/sec^2	ft/sec^2	in/sec^2

where 1 bar = 10^6 dynes/cm^2 = 10^5 N/m^2 = 10^5 Pa
1 atm = 1.013 bars = 14.7 psi = 2116 lb_f/ft^2 = 9.807×10^4 Pa
1 slug = 1 lb_f-s^2/ft = 14.59 kg
1 snail = 1 lb_f-s^2/in
1 gravity = 9.81 m/s^2 = 981 cm/s^2

LIST OF INPUT COMMANDS

Apply

XF	value	<long> <VAR=xv,yv>	i=i1,i2 j=j1,j2
YF		<short>	<from i1,j1 to i2,j2>
PRESSURE		<short>	
SXX			
SYY			
SXY			

The **APPLY** command is used to apply forces, pressures or stresses to any boundary of the model grid. Thermal and groundwater boundary conditions are also applied with this command. (See the appropriate supplementary appendices.) The user must specify the quantity type to be applied (i.e., XFORCE, YFORCE, PRESSURE, SXX, SYY, SXY), the numerical values to be applied, and the gridpoint range over which the variable is to be applied. Optionally, the user may define a variable distribution or a boundary path over which the quantity is to be applied. Some simple rules, listed below, must be followed.

Apply – 2

1. Range Over Which Variables Are Applied

The range over which the variable is applied is defined by the term i=i1,i2 j=j1,j2, where i1,j1 is the starting gridpoint and i2,j2 is the ending gridpoint of the range of boundary points over which the variable is to be applied. Optionally, the range may be given as "from i1,j1 to i2,j2". In either case, both points must be located on a boundary.

Note that the code no longer requires that gridpoints be ordered in a given fashion to produce tensile or compressive pressures on boundaries.

Apply — 3

2. Sign Conventions

Pressure is given as a positive value for pressure acting toward the solid material. Pressure is given as a negative value for pressure acting away from the solid material. Note that this is a change from pre-2.10 versions of FLAC, which required that gridpoints be numbered in a specific order to apply compressive or tensile pressures. The sign convention for applied stresses to boundaries is the same as if they were applied internally—i.e., compressive stresses are negative and tensile stresses are positive. Shear stresses may be positive or negative as determined for the particular stress state. The sign convention for forces is determined from the direction in which the force acts—e.g., a y-force acting in the +y direction is positive; a y-force acting in the -y direction is negative.

Apply — 4

3. Use of a Boundary Path

As described in Rule 1, above, variables may be applied over specified ranges of gridpoints by using the i=i1,i2 j=j1,j2 range descriptors. This is a simple operation if the boundaries are horizontal or vertical, as most often occurs on the outer boundaries of the grid. It is quite difficult to apply variables in this way if the boundary is curved or irregular, such as may be found with a borehole or excavation. For this reason, the code is provided with logic for tracing a boundary path between two specified gridpoints. Three options for defining the boundary path along two boundary gridpoints are provided: LONG, SHORT or BOTH. These options are used in combination with the "from i1,j1 to i2,j2" range descriptor which defines the boundary gridpoints on the path. LONG specifies the longest path between these two points, SHORT specifies the shortest path, and BOTH adds the long and short paths. The default path is the SHORTest distance between the two points. NOTE that, if i1,j1 and i2,j2 are the same, the entire closed boundary will be taken. Examples of the use of this command are given on the following pages.

Apply —5

4. Applying Gradients of Pressure, Force or Stress

In many cases, a linearly variable pressure distribution is required along a boundary. This arises, for example, when inducing far-field stresses through application of boundary stresses. The VARiation XV,YV modifier is used for this purpose. The original value specified by the **APPLY** command is forced to vary linearly from its original value, **VALUE**, to **VALUE + XVAR** or **VALUE + YVAR** from the bottom to top of the i,j range specified. Note that the range i1,j1 i2,j2 refers to gridpoints. An example will help to illustrate the use of this option.

---

*This is a change form pre-2.10 versions. In earlier versions, i1,j1 and i2,j2 referred to zones, and the variations were taken between zone centroids.

Apply — 6

4. Applying Gradients of Pressure, Force or Stress (continued)

The command

apply press=20 var 0,-20 i=1 j=1,11

will cause the horizontal compressive pressure along line i=1 to vary from 20 at gridpoint (1,1) to 0 at gridpoint (1,11). Further examples are given below.

If you want gradients for a closed boundary path, then the starting gridpoint should be different from the ending gridpoint of the range so that the length of the span over which XV and YV are applied is not zero.

Apply —7

EXAMPLE 1: Pressure Applied Around Exterior Boundaries

In this example, compressive pressures are applied along the vertical and horizontal boundaries of the grid so that horizontal pressure = 20 and vertical pressure = 10.

```
grid 10,10
m e
* left vertical boundary
apply press=20 i=1,j=1,11
*right vertical boundary
apply press=20 i=11,j=1,11
*top boundary
apply press=10 i=1,11,j=11
*bottom boundary
apply press=10 i=1,11,j=1
*plot and print results
plot apply
print apply
```

Apply — 8

EXAMPLE 2: Pressure Applied to Exterior Boundaries, With Depth Gradient on Horizontal Pressure

This example is the same as Example 1 but, here, the horizontal pressure is varied from 0 at the top of the grid to the full value at the bottom.

```
grid 10,10
m e
* left vertical boundary
apply press=20 var 0,-20 i=1 j=1,11
* right vertical boundary
apply press=20 var 0,-20 i=11 j=1,11
*top boundary
apply press=10  i=1,11,j=11
*bottom boundary
apply press=10  i=1,11,j=1
plot apply
print apply
```

Apply — 9

EXAMPLE 3: Apply Compressive Stress Field to Exterior Boundaries Which Will Be In Initial Equilibrium with Initial Stress within Grid

```
grid 10,10
m e
*left vertical boundary (compression
*negative)
apply sxx=-20 i=1,j=1,11
*right vertical boundary
apply sxx=-20 i=11, j=1,11
*top boundary
apply syy=-10 i=1,11 j=11
*bottom boundary
apply syy=-10 i=1,11 j=1
*initial stresses in grid
ini sxx=-20 syy=-10
*grid in equilibrium at 2:1 stress field
```

Apply — 10

EXAMPLE 4: Apply Internal Pressure to Tunnel

This example illustrates how pressures may be applied to the surface of a tunnel using the boundary path logic.

```
grid 10,10
m e
gen circ 5,5 3
plot grid
*find gridpoints which lie on boundary
print mark
*note that boundary gridpoints are
*marked
*choose points on boundary between
*which we apply pressures
apply pres=10 short from 6,3 to 4,9
apply pres=30 long from 6,3 to 4,9
plot apply
*note that the plot apply plots grid-
*point forces.  Therefore even for
*constant pressure distribution, forces
*at gridpoints may vary due to zone
*face length.
```

ATtach

Aside <long> from i1,j1 to i2,j2/Bside <long> from i3,j3 to i4,j4

The **ATTACH** command allows the user to rigidly attach pairs of gridpoints to one another along two specified boundaries. This command is useful for creation of grids which have a linear radial expansion of element size. Examples are given below.

The pairs of gridpoints to be attached must be given in order. Referring to the above command nomenclature, (i1,j1) will be attached to (i3,j3) and so on in order of appearance. Gridpoints may be attached which are not touching, but a warning message will result. The shortest boundary path between two points is taken (for closed boundaries) by default, but the longer path may be selected by specifying the LONG keyword. There is no limit to the number of gridpoints that can be attached to one another; however, a single pair of gridpoints cannot be attached. If the ranges (from i1,j1 to i2,j2) of two or more **ATTACH** commands overlap, then groups of multiple attached gridpoints can be created. If the **FIX** and **FREE** commands are applied to a gridpoint, they are also applied automatically to all attached gridpoints. The **PRINT ATTACH** and **PLOT ATTACH** commands can be

ATTach —2

used to verify which points are attached to one another.

A trivial example of the **ATTACH** command is to simply put two halves of a grid together and bond them:

```
grid 10,10
m e      i=1,3
m e      i=5,10
* leaves one row of zones null, creating two
* separate grids. Now, move upper block down
* to meet lower using INITIAL command
ini y add -1 reg=1,10
* attach gridpoints along boundary
att aside from 1,4 to 11,4/bside from 1,5 to 11,5
print attach
```

The two halves of the grid are now attached and will act as a continuum.

It is often desirable, particularly in underground excavation problems, to create grids which are finely discretized in the center of the grid but which grade out to "infinite" boundaries in a radial expansion at some constant ratio. The excavations may then be formed in the central, finely discretized area where stress gradients are high. The **ATTACH** and **GEN** commands allow the user to create this type of grid by forming two grids: one to represent the central, finely dis-

ATTach — 3

-cretized area, and one to represent the outer, expanded portion which grades the mesh to infinite boundaries. Examine the following example problem::

```
* example use of GEN and ATTACH to develop
* a radially-expanding grid about a finely-
* discretized center
grid 52,10
* for a 10x10 central, finely-discretized grid with
* exterior grid "wrapped" around it, create two
* separate grids
m e  i=1,40 j=1,8
* this portion to be wrapped around
m e  i=42,51 j=1,10
* this portion to be the central part
* wrap first portion of grid around itself to get
* radial expansion in ratio of 1:1.2. forms
* rectangular torous. wrap one quadrant at a
* time. each quadrant is 10 elements wide x 8
* elements high.  increase dimensions of outer
boundary of grid to 100 x 100, bottom first
gen 0,0 40,40 60,40 100,0  rat 1.0, 0.83  i=1,11 j=1,9
* right-hand side
gen  s  s  60,60 100,100  rat 1.0,0.83 i=11,21 j=1,9
* top
gen  s s  40,60 0,100  rat 1.0,0.83 i=21,31 j=1,9
* left-hand side
gen  s  s  40,40 0,0  rat 1.0,0.83 i=31,41 j=1,9
```

ATtach — 4

```
* attach the ends of "wrap-around" so that they act as
* one grid
attach a from 1,1 to 1,9 / b from 41,1 to 41,9
* we now have a square with a square hole in the
* middle of dimensions 20 x 20; fill it in with the second
* grid, changing dimensions; gridpoints must coincide
* (mandatory)
gen 40,40 40,60 60,60 60,40  i=42,52    j=1,11
* attach the two grids together on their boundaries; do
* this in two steps, bottom first, then 3 sides at once
attach a from 1,9 to 11,9 / b from 42,1 to 52,1
attach a from 12,9 to 41,9 / b (long) from 52,2 to 42,1
plot grid
print attach
```

A word of caution regarding the **ATTACH** command and "wrapping" grids. When you print values of grid variables (e.g., x, sxx, ydis, etc.), the elements and gridpoints may no longer appear in the same geometrical position as they are in the screen plots. This is due to the fact that the grid has been distorted and no longer conforms to the standard i-j plot representation. Therefore, when obtaining numerical values for stresses, displacements, etc., you must make certain that the location of the zone or gridpoint is known ahead of time.

Call filename

A remote input file, filename, can be run with the **CALL** command. Any series of input instructions can be placed in this file to run in a remote or batch mode. The command **RETURN** must be the last command in the remote file to return input to the local or interactive mode. At present, the file must not contain a **CALL** command itself.

CConfig option

This command allows the user to specify if certain options are desired. The command alters the use of memory in FLAC. The **CONFIG** command only reserves space for the desired options; they need not subsequently be used, but the amount of memory available for other things will be less. If **CONFIG** is used, it must be given before the grid is specified. The options specified through this command are:

(1) groundwater;

(2) thermal;+

(3) axisymmetry;

(4) plane stress; and

(5) z-direction.

The default is to assume that none of these options is to be used (i.e., a plane-strain mechanical problem is solved).

+The thermal configuration can only be used with versions of the program with the thermal option.

Config — 2

The last of these is used to indicate that the out-of-plane stress component must be calculated during mechanical cycling. This option must be specified for the creep models, but may also be specified for others. In the Mohr-Coulomb model, the out-of-plane component may affect the plastic yield. If the out-of-plane component is specified, SZZ may be **INITIAL**ized, **PRINT**ed, **PLOT**ted, or used in a **HISTORY**. For axisymmetry, the z-direction is enabled automatically.

Only one **CONFIG** command is allowed, although more than one option may be requested with it (e.g., **CONFIG AX THER** specifies both axisymmetry and thermal options). If you always will require certain options, it is best to put the **CONFIG** command in the "FLAC.INI" file, so that FLAC is automatically configured correctly.

The keywords for this command are, GW, TH, AXI, P_STR, and Z_D.

See the supplementary notes provided with the manual for information on the solution of axisymmetric problems.

CYcle n

CYCLE is a synonym for **STEP**, where n is the number of calculation steps to execute.

Fix keyword <mark> <range>

With this command, the quantities Xvelocity, Yvelocity, Pore-pressures and Temperatures may be prevented from changin. One or more of the keywords X, Y, PP or T can be used. <range> is of the form I=i1,i2, J=j1,j2; either I or J can be first. If a fixed displacement is desired, the appropriate velocities should be initialized to zero (This is default on start-up.). The **FIX** and **INI XV, YV** commands may be used together to provide a rigid, moving boundary condition. Note that, if the MARK keyword is given, only marked gridpoints in the range will be fixed.

FRee <X> <Y> <PY> <Tem> <Mark> <range>

This is the inverse of the **FIX** command. It releases the gridpoint constraint.

GEn x1,y1 x2,y2 x3,y3 x4,y4 <ratio ri rj> <range> <same> <same> <same> <same>

Coordinates can be generated for the whole grid or for sub-regions. If the above form of the **GEN** command is given, a quadrilateral region (determined by the range of rows and columns) is re-zoned. The gridpoint with the lowest row and column number in the range is placed at coordinate (x1,y1); the remaining corners of the range are given the coordinates (x2,y2), (x3,y3), (x4,y4) proceeding in a clockwise direction. Any or all of the coordinate pairs may be substituted with the word SAME: in this case, the particular corner will retain its current coordinate. The remaining gridpoints in the quadrilateral region are spaced uniformly within the region unless the optional RATIO keyword is given. In this case, the zone spacing can be made to increase or decrease according to the parameters ri and rj, which are the geometric ratios between successive zone sizes in the I and J directions, respectively. For example, a ratio of 1.1 will cause each successive zone to be 1.1 times larger than its predecessor.

GEn — 2

GEn

Circle xc yc rad
Arc xc,yc xb,yb theta
Line x1,y1 x2,y2
Table n

This form of the **GEN** command forces specified portions of the grid to conform to given shapes such as circles, lines, etc. These **GEN** commands are used to define internal shapes within the grid which may be used to define excavations or structures such as bedded deposits or ore bodies. The following shapes are available.

ARC — The grid is mapped to conform to an arc of a circle which is centered at (xc,yc) with a beginning boundary point of xb,yb and a counterclockwise angle of theta degrees.

CIRCLE — The grid is mapped to a circle of radius rad and center (xc,yc).

LINE — The grid is mapped to a line with end coordinates (x1,y1) and (x2,y2).

GEn — 3

TABLE Gridpoints are moved to conform to the coordinate pairs specified in table number n (See **TABLE** command). Note that this command is used to form a closed path. The first and last points must be the same.

Note that the gridpoints that form part of the given shapes are "marked" (see commands **MARK** and **UNMARK**).

The whole region enclosed by marked gridpoints can then be addressed by the **REGION** keyword with the commands **INITIAL**, **MODEL** and **PROPERTY**. For example, the region within a circle can be removed with the command **mod null reg=i,j**, where i,j is one zone inside the circle. Examples of the use of the **GEN** command are given in Section 5.2.1.

The **GEN** command should be used with some caution. Some geometries are impossible to construct with quadrilateral elements (e.g., long, thin triangular regions defined by two intersecting lines). Also, the user should be aware that once gridpoints are MARKED, they cannot be moved again by another **GEN** command. If the user wishes to move a

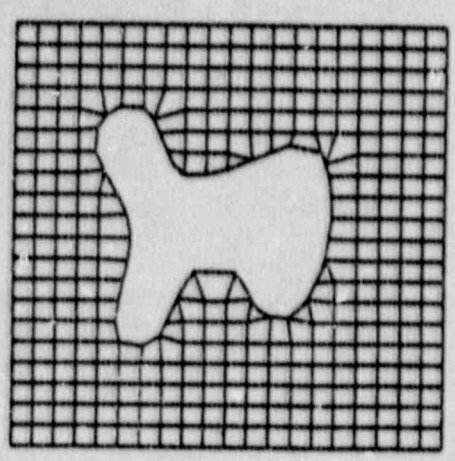
GEn — 4

gridpoint which was MARKED previously, the gridpoint must first be UNMARKED.

The following example illustrates the use of the **GEN TABLE** command:

```
g 20 20
m e
ini x mul 10 y mul 10
tab 1 59,47 50,50 48,60 50,72 55,95 55,104 52,114 38,138 38,150
tab 1 44,159 54,163 63,160 75,133 82,128 100,132 131,145 142,144
tab 1 147,130 149,113 145,87 136,68 127,61 116,64 104,85 95,90
tab 1 86,86 78,75 69,53 59,47
gen tab 1
mod null reg=10,10
plot grid
```

The resulting grid is shown below.



GEn --- 5

GEn ADJust

The grid adjusts itself in an attempt to obtain a smoother discretization. Grid-points that are marked and boundary points are not moved. Successive **ADJUST** commands may be given to refine the grid further. Examples of the generation of grids can be found in Section 5.2.1

Grid icol jrow

The number of columns and rows of zones in the calculation grid is specified. For the 640K version of FLAC, approximately 2,000 zones may be used, with the Mohr-Coulomb model. The 2 mb and 4 mb 80386 versions allow about 4,000 and 15,000 zones, respectively.

Help The HELP command provides a screen listing of the available FLAC commands.

His <Nstep=n> <keyword . . . I=i1 J=j1>

The history of a variable <keyword> is stored every NSTEP timesteps for gridpoint or zone i1,j1. Up to 1,000 points per history may be kept for up to 25 histories at any time. Each history is numbered sequentially from 1. The histories requested may be printed at any time by using the **HIS LIST** command. NSTEP must be the same for all histories and need only be given once prior to timestepping; it defaults to 10 if not given.

The history values are stored on a direct access file on the hard disk. This file is erased when stopping FLAC; therefore, if the user wishes to save the history, a save file must be created or the history must be written to a formatted disk file (see **HIS WRITE N** command). The keywords for which histories may be tracked for gridpoint or zone i,j are:

Ang	angle of minor principal stress counterclockwise from positive x-axis for zone i,j
ASXX	xx-stress, triangle a of zone i,j

His — 2

ASXY	xy-stress, triangle a of zone i,j
ASYY	yy-stress, triangle a of zone i,j
ASZZ	zz-stress, triangle a of zone i,j
BSXX	xx-stress, triangle b of zone i,j
BSXY	xy-stress, triangle b of zone i,j
BSYY	yy-stress, triangle b of zone i,j
BSZZ	zz-stress, triangle b of zone i,j
CSXX	xx-stress, triangle c of zone i,j
CSXY	xy-stress, triangle c of zone i,j
CSYY	yy-stress, triangle c of zone i,j
CSZZ	zz-stress, triangle c of zone i,j
DSXX	xx-stress, triangle d of zone i,j

His — 3

DSXY	xy-stress, triangle d of zone i,j
DSYY	yy-stress, triangle d of zone i,j
DSZZ	zz-stress, triangle d of zone i,j
PP	pore pressure for zone i,j
SIG1	major principal stress, zone i,j
SIG2	minor principal stress, zone i,j
SXX	xx-stress, zone i,j
SYY	yy-stress, zone i,j
SXY	xy-stress, zone i,j
SZZ	zz-stress, zone i,j
Temp	temperature of gridpoint i,j
Unbal	maximum unbalanced force
X	x-coordinate of gridpoint i,j
Y	y-coordinate of gridpoint i,j

His —4

XDis	x-displacement, gridpoint i,j
YDis	y-displacement, gridpoint i,j
XVel	x-velocity of gridpoint i,j
YVel	y-velocity of gridpoint i,j

His — 5

His keyword

The following forms of the **HISTORY** command allow the user to print, write or reset histories. History number nhis (nhis=1 to total number of histories) is written to the screen or to a disk file on the hard disk. The following choices of the keyword are possible.

Dump nhis	The history (time-step number, history value) of history number nhis is written to the screen. If the history is greater than the page length, it will scroll past the screen. Use the cntrl-numlock keys to stop scrolling.
List	lists the histories currently being sampled.

His —6

Write nhis	The history (time-step number, history value) of history number nhis is written to a file FLAC.HIS on the hard disk. This file may be printed or manipulated after stopping FLAC. Successive **HIS WRITE** commands will sequentially add to the FLAC.HIS file. However, the first file written will overwrite any existing FLAC.HIS file.
Reset	All histories are cleared.

Initial

<mark> var_name <add> value <var xv,yv> <. . .> <range> <mul>

Certain gridpoint or zone variables are given initial values. One or more keywords can be chosen from the following list:

PP	pore pressure
X	x-coordinate
Y	y-coordinate
SXX	xx-component of stress+
SYY	yy-component of stress+
SXY	xy-component of stress+
SZZ	zz-component of stress

+Remember that compressive stresses are negative!

Initial — 2

XDisp	x-displacement
YDisp	y-displacement
XVel	x velocity*
YVel	y velocity*

Several optional keywords are provided to assist implementation of the **INITIAL** command. These are:

Add	The keyword ADD adds the specified value (including any variations given) to existing values (of the variable specified) in the grid. This command is useful, for example, in translating coordinates for regions of a grid without resorting to the **GEN** command, as is encountered when creating interfaces.
Mark	Gridpoint variables will be initialized on MARKed gridpoints only (others will remain unaffected).

*Velocity units are length per timestep.

Initial — 3

Mul

The keyword MUL multiplies the existing values of the specified variable by the quantity given (including any variations). This command is useful in distorting the grid, changing stress gradients, etc. Do not use negative multipliers unless you really know what you are doing.

Region I J

All zones enclosed by a continuous sequence of MARKed gridpoints will be initialized. The zone I,J is one zone within the MARKed region.

Var xv,yv

A variation in the parameter can be given over the specified range (or region) where xv is the variation in the x-direction and yv is the variation in the y-direction. The variations are applied between the minimum and

Initial —4

Var (continued)

maximum x and y values in the i,j range specified. This is a change from pre-2.10 versions of FLAC. When placing initial stresses in the grid, the stress variation or gradient is applied between gridpoints, not zone centroids, as in previous FLAC versions.

An optional range of zones or gridpoints can be given, identified by <range>. The range, of the form I=i1,i2, J=j1,j2, must be last on the input line, but either I or J can be first.

Initial — 5

Example 1: Initial Horizontal Stress Field Varying With Depth

If you wish to initialize a gradient in the horizontal stress with depth, the following **INITIAL** command may be issued:

```
grid 10,10
m e
ini sxx=-30e6 (var=0, 10e6) i=1,11 j=1,11
fix x y i=1
fix x y i=11
fix x y j=1
fix x y j=11
solve
print sxx
```

Here, the xx-stress is -30 MPa at the bottom of the grid and is set to -20 MPa at the top of the grid, with a linear variation in y between these gridpoints.

Initial — 6

Example 2: Translation of a Portion of a Grid to Form a Diagonal Interface Using the **INITIAL** Command

```
gr 5,20
m e
*generates interface
gen line (0,3) (5,14)
gen line (0,5) (5,16)
mod null reg=1,5
*shift top block down
ini y add -2 reg=1,20
*re-zone endpoints
ini x=5.0,y=14.0 i=6,j=14
ini x=0.0,y=3.0 i=1,j=8
```

The **INITIAL** command can be used to move gridpoints to other locations in the grid. Care must be taken when performing this procedure. FLAC internally divides the quadrilateral zones into two superimposed pairs of triangular zones for the calculations (see Fig. 3-6).

Initial —7

A quadrilateral may be deformed in any fashion, subject to the following two criteria:

(1) the area of the quadrilateral must be positive; and

(2) each member of at least one pair of triangular zones must have an area greater than 20% of the total quadrilateral area.

If either of these criteria is not met, FLAC will give a "BAD GEOMETRY" error message during timestepping. Figure 4-2 illustrates possible zone deformations.

It is often useful to initialize the displacements prior to a run by setting their values to zero. This will not affect the calculations and will give the change in displacements as output. If they are not initialized, the displacement upon output will be the total displacement.

CAUTION: It is not meaningful to change coordinates part-way through the run. The user may specify an initial stress state which may not be in equilibrium—in which case, the model should be equilibrated first.

Initial — 8

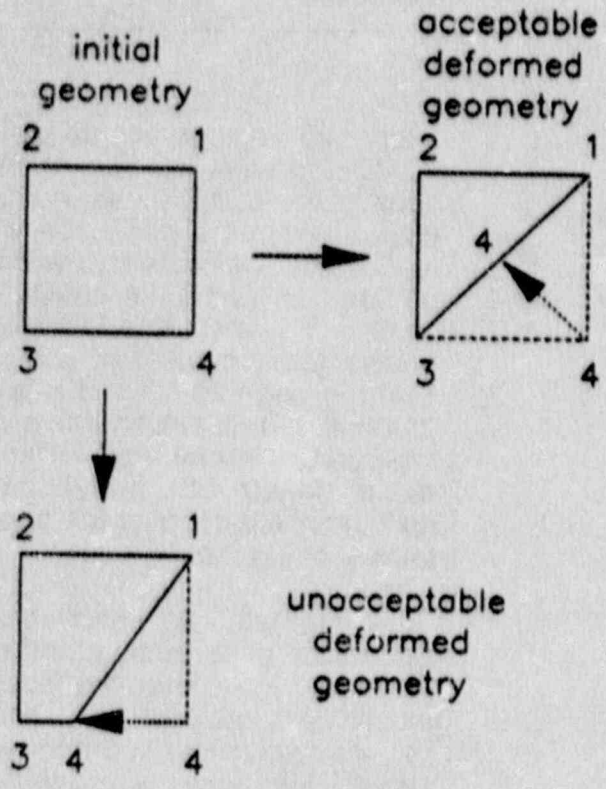
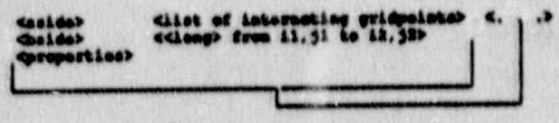


Fig. 4-2 Acceptable and Unacceptable Zone Deformations

INTerface



Interfaces or joints can be defined in a FLAC grid between two or more portions of the grid. These interfaces are planes within the grid upon which slip and/or separation is allowed and can be used to model the effects of joints, faults, frictional interfaces between bodies (e.g., cement in a bin, a billet being forced through a die, a concrete foundation on a soil mass), etc. Within the code, the interface is represented as a series of normal and shear springs which connect the opposing interface at interacting nodes.

The interface is given shear and normal stiffness, friction, and cohesion. Additionally, the interface may be given tensile strength and may be glued—i.e., the two halves are bonded and no slip or separation is allowed. In any case, KN and KS must still be given.

The following keywords and options can be used with the **INTERFACE** command.

INTerface — 2

KEYWORD	VALUE	UNITS	OPTIONAL
Aside	from i1, [illegible] i2		i1, j1, i2, j2, ..., in, jn
Bside	from i1, j1, [illegible] i2		
Cohesion	value	(stress)	
Friction	value	(degrees)	
KN	value	(stress/length)	
KS	value	(stress/length)	
Tbond	value	(stress)	
Glued			
Unglued			

The value n is a unique identification number for the interface and need not be sequential.

Figure 4-3 shows a typical interface and the nomenclature used for describing it.

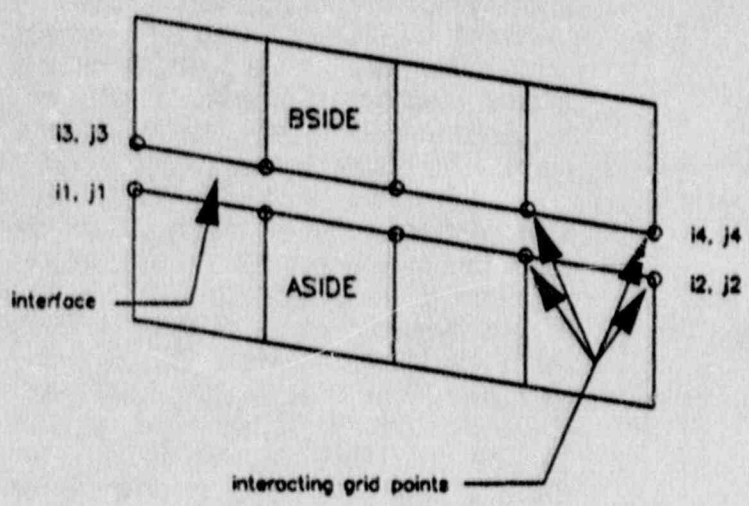


Fig. 4-3 Interface Nomenclature

INTerface — 3

The two interacting faces of the interface are termed the ASIDE and the BSIDE. These faces are initially created by nulling the region between them and by defining the possible interacting gridpoints. The two faces are then generally placed in contact using the **GEN** or **INITIAL** command prior to timestepping. The previous versions of FLAC required that the user specify the interacting gridpoints. The code now does this automatically by simply specifying the end gridpoints along each side of the interface using the FROM... TO... construction. If this form is used, the shortest distance between the specified gridpoints will be taken; use the keyword **LONG** to force the longer route to be taken. The present version allows intersecting interfaces. Appendix D provides greater detail on the generation of interfaces.

The interface must be given properties prior to timestepping. These include the normal and shear stiffness (kn, ks) in stress/length (e.g., GPa/m, psi/in), the cohesion in stress units (i.e., Pa, psi), and friction angle (degrees). Additionally, the tensile strength (stress) can be specified, as well as whether the interface is to be glued (no slip or separation) or, conversely, unglued.

INTerface — 4

Any set of consistent units may be used for these properties. If the properties are not specified, they are assumed to be zero.

Example problems for use of the **INTERFACE** logic are presented in Appendix D.

MArk <range>

Gridpoints are marked in the given range. A marked gridpoint does not influence the calculations in any way but serves to delimit a region for the purposes of the **INITIAL**, **PROP**, and **MODEL** commands. Note that the **GEN** command marks some gridpoints automatically.

<range> is of the form I=i1,i2, J=j1,j2. Either I or J may be first.

Model keyword <REGION=i,j> <range>

This command associates a constitutive model with an area of the grid corresponding to the range of zones identified by <range> or the region in which zone i,j lies. See Section 5.2.1 for greater detail on defining regions within a FLAC grid. During the calculation cycle, zones will refer to a constitutive subroutine corresponding to one of the keywords listed below:

ANisotropic
: transversely isotropic elastic model

Elastic
: elastic, isotropic constitutive model

Mohr-Coulomb
: Mohr-Coulomb plasticity model

Null
: null zone (Null zones are used to model excavated material. The stresses within the null zone are automatically set to zero.)

Model — 2

SS	strain softening
Ubiquitous	ubiquitous joint model

<range> is of the form I=i1,i2, J=j1,j2; either i or j can be first. Alternatively, a region of the model can be defined as REGION=i,j; in this case, all zones enclosed by a continuous sequence of **MARKED** gridpoints will be set to the given model.

The **MODEL** command is also used to specify thermal and creep models, where appropriate. (See supplementary appendices for details.)

New This command allows the user to begin a new problem without leaving FLAC.

PLot keyword <switch<= v>...><keyword ...>

This command requests a plot to be made on the screen, Hewlett-Packard pen plotter or Adobe PostScript-compatible laser printer, or allows HP or PostScript command files to be sent to disk. Plots can also be sent to a file in a format to be used by dot matrix printers. Note that the **WINDOW** command must first be used to define the plotting area unless the default window is satisfactory. Several variables may be plotted at once by giving several keywords on one line. The grid must be given a material model (using the **MODEL** command) prior to making printouts or plots. The keywords, and their meanings, are as follows:

Apply	plots vectors of applied force resulting from the **APPLY** command.
Attach	Attached gridpoints are indicated.
BEam	plots geometry of structural elements.
Boundary	The boundary of the grid is plotted, including internal boundaries of null regions.

PLot — 2

Cable	plots cable bolt geometry.
CForce	plots force vectors along cable bolt elements.
Disp	Displacement vectors are plotted.
E_p	contours of plastic strain (if using strain-softening material model only)
Fix	Fixed gridpoints are shown.
Flo	Flow vectors are plotted.
GNum	Gridpoint i,j numbers are shown.
Grid	The finite-difference grid is plotted.
Head	hydraulic head contours

PLot — 3

His n — A plot of the time history of the variable recorded in history number n is made. This keyword can also be used to show (e.g., on a grid plot) where histories are taken.

The command **PLOT HIS** n will plot history n, but **PLOT GRI HIS n** will show the location of history number n on a grid plot. **PLOT GRI HIS** will show the locations of all histories.

Hoek — This command is similar to the **PLOT MOHR** command. The Hoek-Brown empirical strength parameters ucs, m, s must be previously-defined using the **SET** command

Mark — Marked gridpoints are shown.

PLot — 4

Model	creates a filled grid plot of the various material models in the grid (screen or PostScript plot only).
Mohr	allows the user to plot contours of strength/stress ratios for zones with a Mohr-Coulomb constitutive relation. The cohesion and friction angle used for this plot are input using the **SET** command and are not necessarily the same as those in the zones being plotted.
Num	Zone numbers are shown on the plot.

PLot — 5

Pen routes screen plot tc one of two locations: (1) a Hewlett-Packard 7470A or 7475A (or compatible) pen plotter or PostScript-compatible laser printer; and (2) a disk file (The disk file will contain HPGL or PostScript commands or commands for a dot matrix printer, depending on which printer type is set (see the **SET** command)).

These files may be copied, after quitting FLAC, to the pen plotter or laser printer using the **DOS** copy or print commands.

If a dot matrix printer is to be used, see Appendix G.

The plot routing is controlled by three **SET** commands which are given prior to issuing a **PLOT** command.

PLot — 6

Pen (continued)

(a) **set out = name**

where **name** is the disk filename to be created for diversion to a disk file or is the output port (com1, com2, or LPT1) to which the printer or plotter is connected;

(b) **set baud = baud**

which sets the **baud** rate of the output port (baud = 1200, 4800 or 9600); and

(c) **set plot = device**

where **device** = HP (Hewlett-Packard pen plotter); or **device** = POST, for PostScript laser printer file, or **device** = PRINTER for dot matrix printer.

PLot — 7

Pen (continued)

To create pen plots: (1) **set** the **baud** rate; (2) **set output** (file or port); and (3) **set plot** (HP or POST).

To send plots, type **plot pen type**, where **type** is any plot type. The keyword **pen** must appear immediately after the **PLOT** command.

PLAS — plots a symbol in each zone indicating whether yield has occurred.

PP — contours of pore pressure

Prop_name

(where prop_name is any material property for the present grid) This creates a filled grid plot of the properties in the grid. Example: **plot shear** will create a plot of all shear moduli in the grid.

PLot — 8

Region	creates a filled grid plot of the various regions within the grid (screen or PostScript plot only).
RF	reaction forces on fixed gridpoints, plotted as vectors
Sig1	contour plot of major principal stresses
Sig2	contour plot of minor principal stresses
STAte	The current yield state of the material is plotted at element centroids.
STress	principal stress vectors
SXX	contours of xx-stress (total stress)
SYY	contours of yy-stress (total stress)

PLot — 9

SXY	contours of xy-stress
SZZ	contours of zz-stress
Table n	Table n is plotted.
Temp	temperature contours
Tension	allows the user to plot a single contour separating regions of tension and compression.
Theta	angle between principal stress components
Velocity	Velocity vectors are plotted (as arrows).
WAter	plots water table line.*

*Note: You must re-adjust window if water table lies above the ground surface as in an undersea simulation.

PLot — 10

XDisp	contours of x-displacement
YDisp	contours of y-displacement
XVel	contours of x-velocity
YVel	contours of y-velocity

PLot — 11

Enhancing or Modifying Plots

If no further parameters are given, scale factors, a window and a default line color will be chosen automatically. However, any keyword may be followed by any number of "switches", which are themselves keywords that set certain characteristics of the plot. Each switch operates on the keyword that precedes it. The switches can be used to control pen color, scaling of the vectors, contour interval, drawing of geometric scales on the plot boundaries, drawing of profile or line plots of a variable, and removal of headings.

Note that an enlarged or contracted plot may be made by giving a **WINDOW** command prior to giving the **PLOT** command. The window will then remain set and will be remembered on restart.

ABSO	The magnitude of the quantity is plotted.
BLO	Instead of contour plots, zones are filled with a different color for each value of the property.

PLot — 12

Enhancing or Modifying Plots
(continued)

Color	The line color desired is typed literally on the command line (e.g., **plot grid red**). Color choices depend on the palette chosen previously with the **SET PAL** command and the type of graphics monitor.* For the CGA palette #0, the line color choices are: Green, Red and Yellow-brown; for palette #1, the line color choices are Cyan, Magenta and White.

*Note that the background color of the plot may be adjusted. For CGA monitors, the background normally is set to black (i.e., **set bac=0**; for the monochrome mode, use **set bac=2**). The default palette choice is pal=1.

PLot — 13

Enhancing or Modifying Plots
(continued)

EGA Color Choices

Line Color	Keyword
green (default)	GREen
red	Red
yellow	Yellow
magenta	MAGenta
brown	Brown
white	White
light green	LGreen
light red	LRed
light blue	LBlue
light magenta	LMagenta
gray	GRAy
intense white	IWhite
black	Black
blue	BLue
cyan	CYan

PLot — 14

Enhancing or Modifying Plots
(continued)

Changes have been made for the choice of pen color for H-P six-pen plotters. The choice of pen number has been set up to use all six pens in the H-P pen plotter. The pens chosen correpond to the screen colors in the EGA and CGA modes. The following describes the pen assignments for the various color choices.

In EGA Mode

Screen Color	Pen Selection
green	1
cyan	1
red	2
blue	3
magenta	4
brown	5
white	6
gray	1
light blue	2
light green	3
light cyan	4
light red	5
light magenta	6
yellow	1
black	2

PLot — 15

Enhancing or Modifying Plots
(continued)

In CGA Mode

Screen Color	Pen Selection
green	1
cyan	1
red	2
magenta	2
yellow	3
white	3

If you have only a CGA adaptor, you may still plot in all six colors by issuing the **SET EGA** command prior to issuing the **PLOT PEN** ___ commands. Use the **SET CGA** command prior to making any screen plots.

PLot — 16

Enhancing or Modifying Plots
(continued)

Fill	is used to create filled contour plots, as opposed to the default line contour plots (for PostScript or screen plots only).
Interval = c	sets the contour interval to c. The contour values may be written on the screen or pen plot by issuing the **SCLIN** command immediately before issuing the **PLOT** command (see **SCLIN** command).
INV	For PostScript plots, the gray-scale is reversed for filled plots.

PLot — 17

Enhancing or Modifying Plots
(continued)

Line xb,yb xe,ye,num

creates an x-y type plot of gridpoint or zone variable versus distance along the line defined by the endpoints (xb,yb) (xe,ye). The line is divided into num points, and the zone or gridpoint variables are interpolated at each point from surrounding zone centroids or gridpoints. Care should be taken when specifying the value of num. It is not sensible to specify num such that the spacing of points along the line is much less than the gridpoint spacing.

PLot — 18

Enhancing or Modifying Plots
(continued)

Max = v	sets the value corresponding to the maximum length of arrow to v for vector plots. All other vectors will be scaled to this maximum length. On contour plots, the maximum contour value can be set.
Min=v	sets the minimum value to be used on contour plots.
Noh	removes the heading from the plot.
Scales	places geometric scales along the edges of the screen plot (Scales are plotted on pen plots by default.)

PLot — 19

Enhancing or Modifying Plots
(continued)

Time	Histories are plotted as a function of time rather than timestep. This only applies to creep runs.
Zero	causes the zero contour line to be omitted.

Print keyword <keyword> . . . <Region=i,j> <range>

Printed output is produced for applied loads, limits, main grid variables, etc. Output can be produced for a range of gridpoints or zones identified by <range> or the region in which zone i,j lies. <range> is of the form I=i1,i2, J=j1,j2; either i or j can be first. If no range is given, the entire grid is printed. Alternatively, a region of the model can be defined as REGION=i,j; in this case, output associated with zones enclosed by a continuous sequence of MARKED gridpoints will be printed. The zone i,j is one zone within the MARKED region.

As is the case with plotting, the grid variables will not print until a material model is ascribed. The keywords are:

ANgle	prints angle of anisotropy for transversely elastic model.
Apply	values and ranges of forces or pressures applied to the model and thermal and groundwater boundary conditions

Print — 2

ATtach	prints all pairs of attached gridpoints.
Fix	indicates whether x, y, pore pressure or temperature is fixed at gridpoints.
Interface	prints out the interface data, including nodal stresses and unit normal vectors. Note that compressive stresses are negative.
Limits	limits for **SOLVE** command
MArk	prints marked gridpoints.
MEm	memory in use
STRuct	prints relevant forces, moments and displacements for nodes within structural elements
Table	prints table values.

Print — 3

Main Grid Keywords

Bulk	bulk modulus
COhesion	cohesion
Den	mass density
Dilation	dilation angle
E_p	total plastic strain (strain-softening model)
FRiction	friction angle
SHear	shear modulus
X	X-coordinate
Y	Y-coordinate
PP	pore pressures
JAngle	joint angle
JCoh	joint cohesion
JFric	joint friction
JTen	joint tensile strength
NUYx	y-x Poisson's ratio
NUZx	z-x Poisson's ratio

Print — 4

Main Grid Keywords (continued)

Sig1	major principal stress+
Sig2	minor principal stress+
SXx	XX-stress+
SYy	YY-stress+
SXY	XY-stress+
SZZ	ZZ-stress+
TEnsion	tensile strength for Mohr-Coulomb model
THeta	angle of minor principal stress counterclockwise from the x-axis
XDis	X-displacement
YDis	Y-displacement
XMod	X-modulus
YMod	Y-modulus
XVel	X-velocity
YVel	Y-velocity

+Remember, on printouts, compressive stresses are negative.

Print — 5

Main Grid Keywords (continued)

ASXx	XX-stress (triangle A)#
BSXx	XX-stress (triangle B)
CSXx	XX-stress (triangle C)
DSXx	XX-stress (triangle D)
ASYy	YY-stress (triangle A)
BSYy	YY-stress (triangle B)
CSYy	YY-stress (triangle C)
DSYy	YY-stress (triangle D)
ASXY	XY-stress (triangle A)
BSXY	XY-stress (triangle B)
CSXY	XY-stress (triangle C)
DSXY	XY-stress (triangle D)
ASZz	ZZ-stress (triangle A)
BSZz	ZZ-stress (triangle B)
CSZz	ZZ-stress (triangle C)
DSZz	ZZ-stress (triangle D)

#Recall that FLAC divides each quadrilateral zone into four triangular zones: a, b, c and d.

Print — 6

STate	plastic state	
	0	elastic
	1	at yield
	2	elastic but previously at yield
	3	has surpassed tension cut-off in uniaxial tension
	4	at yield, also surpassed tension cut-off
	5	has surpassed tension cut-off in general tension
	6	ubiquitous joints currently at yield
	7	ubiquitous joints have yielded in the past but are currently in the elastic state

Print — 7

XReaction	x-reaction force
YReaction	y-reaction force

Variables specific to the groundwater, thermal and creep models can also be printed. (See the appropriate appendices for details.)

PROp keyword=value <. . .> <region=i,j> <range>

This command assigns properties for a constitutive model identified by the **MODEL** command. The property keywords, their meanings, and the constitutive models which use them are given on the following pages.

Creep, thermal and groundwater flow properties are also assigned. (See the appropriate appendices for details.)

PROp — 2

Keyword	Description	Model
Angle	angle of anisotropy taken counter-clockwise from the x-axis	transversely isotropic
Bulk	elastic bulk modulus	elastic, Mohr-Coulomb strain-softening/hardening, ubiquitous joints
Cohesion	cohesion	Mohr-Coulomb, strain-softening/hardening
CTab,* DTab, FTab	table number	strain-softening/hardening
Density	mass density	all models
Dilation	angle of dilation (in degrees)	Mohr-Coulomb, strain-softening/hardening, ubiquitous joints
Friction	angle of internal friction (in degrees)	Mohr-Coulomb, strain-softening/hardening

*If CTab, DTab or FTab are specified, a corresponding table must be defined (see TABLE command).

PROp — 3

Keyword	Description	Model
JAngle	angle in degrees, counter-clockwise from x-axis for ubiquitous joints	ubiquitous joints
JCOhesion	cohesion of ubiquitous joints	ubiquitous joints
JFriction	friction angle of ubiquitous joint surfaces	ubiquitous joints
NUYx	Poisson's ratio for normal stress in the plane of isotropy due to uniaxial stress in perpendicular plane	transversely isotropic
NUZx	Poisson's ratio for normal isotropic stress in the plane of isotropy due to uniaxial stress in plane of isotropy	transversely isotropic
Jtensile	joint tensile strength	ubiquitous joints
Shear-mod	elastic shear modulus	all models
TEnsion	continuum tensile strength	
Xmod	elastic Young's modulus in the plane of isotropy	transversely isotropic
Ymod	elastic Young's modulus in in the plane perpendicular to isotropy	transversely isotropic

PROp — 4

The model type must be defined prior to printing or plotting of any data. If properties are given which are not consistent with the chosen model, a warning message will be given which informs the user that the unneeded properties were not accepted. The models require the following properties:

Elastic

(1) shear modulus

(2) bulk modulus

(3) density

Mohr-Coulomb

(1) shear modulus

(2) bulk modulus

(3) density

(4) friction angle

(5) cohesion

(6) dilation (optional)

(7) tensile strength (optional)

PROp — 5

Transversely-Isotropic

(1) shear modulus*

(2) x-modulus

(3) y-modulus

(4) density

(5) NUYx

(6) NUYz

---

*The cross shear modulus Gxy for anisotropic elasticity must be determined. S. G. Lekhnittskii (Theory of Elasticity of an Anisotropic Body. Moscow: Mir Publishers, 1981) suggests the following equation based on laboratory testing of rock:

$$G_{xy} = \frac{E_x E_y}{E_x(1 + 2\nu_{xy}) + E_y}$$

assuming the x-z plane is the plane of isotropy (see Appendix A).

PROp — 6

Ubiquitous Joint

(1) shear modulus

(2) bulk modulus

(3) density

(4) cohesion (intact material)

(5) friction (intact material)

(6) jcohesion (joints)

(7) jfriction (joints)

(8) jangle (joints)

(9) dilation (intact material; optional)

(10) jtension - tensile strength of joints (optional)

PROp — 7

Strain-Hardening/Softening

(1) shear modulus

(2) bulk modulus

(3) density

(4) friction angle (initial value)

(5) cohesion (initial value)

(6) dilation (initial value)

(7) ftable (table # for relationship of ϕ to plastic strain, optional)

(8) ctable (table # for relationship of cohesion to plastic strain, optional)

(9) dtable (table # for relationship of dilation angle to plastic strain, optional)

PROp —-8

The strain-hardening and softening behavior is controlled by the variation in friction, cohesion and dilation as a function of plastic strain given by a specified table of values. See Appendix A for details of specifications of these values.

Note that, if table numbers are given as 0, the properties will take the values given (i.e., with COHESION, DILATION or FRICTION keywords).

Optional keywords are provided to assist implementation of the **PROPERTY** command. These are:

Region i,j

All zones enclosed by a continuous sequence of MARKED gridpoints will have the specified properties. The zone i,j is one zone in the MARKED region.

PROp — 9

Var xv yv

A variation in the property can be given over the specified range (or region) where xv is the variation in the x-direction and yv is the variation in the y-direction. The variations are applied linearly between the gridpoints within the specified range, with the smallest and largest i,j numbers. Note that this is a change from pre-2.10 FLAC versions, in which the variation was taken between zone centroids rather than gridpoints.

An optional range of zones can be given, identified by <range>. The range, of the form I=i1,i2, J=j1,j2, must be last on the input line but either i or j can be first. If no region or i,j range is specified, the given property is assigned to every zone in the entire grid.

Quit stops execution of FLAC; a synonym of **STOP**.

REstore filename

A previously-saved problem state is restored from the named file, filename. (See the **SAVE** command.)

RETurn This command should be last in a remote input file (see **CALL** command).

SAve filename

The current problem in memory is saved at its present state on the named file, filename. If the file already exists, it will be overwritten.

SCIIn <n x1,y1 x2,y2>
<reset>

This command labels contours on any contour plot produced on the screen or pen plotter. The user defines a "scan line" upon which he or she wishes to know the values of contours. The user defines the endpoints of this line using the command **SCLIN**. At each point of intersection between the scan line and a contour, a letter, A to Z, is written. A heading is given which indicates the value of the smallest and largest contours and the contour interval. This allows calculation of any contour value between the extremes. The default condition is no scan line. Up to 5 scan lines may be specified for a plot. The parameters are as follows:

n the number of this scan line (must be 1,2,3,4 or 5),

x1,y1 coordinates of the beginning of the scan line, and

x2,y2 coordinates of the ending of the scan line.

The scan lines are remembered between plots. To prevent them from being used, the command **SCLIN RESET** should be used.

SCIin — 2

For example, the command

flac: **sciin 1 0,1 0,10**

will produce values for those contours which intersect the scan line from 0,1 to 0,10.

SEt <keyword> <keyword> . . .

This command is used to set many parameters, most of which control either cycling or plotting. The keywords for this command are:

A3 sets pen plotter paper size to A3.

A4 sets pen plotter paper size to A4 (default).

Aspect a

where a is the ratio of x to y measured from a non-square screen image which should be square. a will generally vary from 0.5 to 1.5, depending on the graphics board and screen vertical size adjustment. You should first attempt adjustment of vertical size control on the monitor.

SEt —2

Back background color

0	black
1	blue
2	green
3	cyan
4	red
5	magenta
6	yellow-brown
7	white

Default assumes that color screen is present.

This command only applies to machines with CGA or monochrome screens.

BAud b

sets baud rate for output device. The choices for b are 1200, 2400, 4800 and 9600 bpm. The baud rate can only be set for COM1 or COM2 and is not used on 80386 versions of FLAC.

SEt —3

CGA or CSC

sets graphics mode to color screen, 320x199 pixel resolution. This is the default screen setting.

Clock t

sets the limit of elapsed time, in minutes, for the **SOLVE** command

Col n

n is the maximum number of columns on output. Note that the user must set the proper column mode on the printer prior to FLAC operation. Consult your printer manual for the method of setting printer modes.

SEt — 4

COLS = npal
= Table ntab
= ncol, lcol
= ncol, color

This keyword is used to control the colors used for filled plots in FLAC.

The first format given above, **SET COLS = npal**, selects one of the three built-in color palettes. (The default palette is palette 1.) The colors in these palettes are tabulated below.

The second format, **SET COLS TABLE = ntab**, uses the colors previously specified by the user with the **TABLE** command. The table must be input as pairs of numbers, the first member of the pair being the location in the table (number from 1 to 13), and the second being the color number corresponding to the desired color on the following list.

SEt — 5

COLS (continued)

For example, the commands

TABLE 3 1,0 2,3 3,5 . . . 13,7

SET COLS=TABLE 3

set up a palette with black, cyan and magenta as the first three colors, and white as the last color.

0 = black
1 = blue
2 = green
3 = cyan
4 = red
5 = magenta
6 = brown
7 = white
8 = gray
9 = light blue
10 = light green
11 = light cyan
12 = light red
13 = light magenta
14 = yellow
15 = intense white

SEt —6

COLS (continued)

The third option, **SET COLS ncol, lcol**, changes color number ncol in the current palette to color lcol on the above list (e.g., **SET COLS 3 7** changes the third color to white).

The fourth option, **SET COLS ncol, color**, acts similarly to the third, except that the color name (black, blue, etc.) is used instead of the color number (e.g., **SET COLS 3 white** is equivalent to **SET COLS 3 7**).

SEt —7

COLS (continued)

Color Tables

Color Table 1

13 = light magenta
12 = light red
4 = red
6 = brown
14 = yellow
10 = light green
2 = green
11 = light cyan
3 = cyan
9 = light blue
1 = blue
7 = white
8 = grey

SEt —8

COLS (continued)

Color Table 2

8 = grav
7 = white
1 = blue
9 = light blue
3 = cyan
11 = light cyan
2 = green
10 = light green
14 = yellow
6 = brown
4 = red
12 = light red
13 = light magenta

SEt —9

COLS (continued)

Color Table 3

1 = blue
2 = green
3 = cyan
4 = red
6 = brown
8 = gray
9 = light blue
10 = light green
11 = light cyan
12 = light red
13 = light magenta
14 = yellow
15 = intense white

SEt — 10

Echo <ON> <OFF>

SET ECHO ON or **SET ECHO** causes FLAC to echo lines coming from a **CALL**ed file to the screen and to the log file (if it is open). This is the default setting. **SET ECHO OFF** prevents the input lines from being displayed.

EGA

sets graphics mode to the enhanced graphics adaptor, 640x350 pixel resolution.

Flow <ON> <OFF>

Causes FLAC to do flow calculation (**FLOW ON**), or suppress flow calculation (**FLOW OFF**). The ON option is the default if **CONFIG GWFLOW** is given.

Force f

sets the out-of-balance force limit for the **SOLVE** command.

SEt —11

Gravity	g <th>
	Gravity is specified by a magnitude, g, and direction angle th, measured from the negative y-axis.
HBM	val
	val is the m coefficient in the Hoek-Brown model. It is used for plotting contours of factor of safety.
HBS	val
	val is the s coefficient in the Hoek-Brown model. It is used for plotting contours of factor of safety.
Large	large strain (coordinates are updated)
Legend	ON/OFF
	If OFF, plots appear on-screen without titles and other data.

SEt — 12

Log ON/OFF

ON opens a file named FLAC.LOG on the default disk drive. If a file FLAC.LOG already exists, it is overwritten. Any text which is printed to the screen from this point on is also written to the log file. This is particularly useful for keeping a record of interactive sessions. The file may be edited to create batch data files.

OFF turns off the logging function. It does not close the log file FLAC.LOG. If set log on is given at some later stage in the session, subsequent screen output will be appended to the file.

SEt —13

Mech	ON/OFF Causes FLAC to do mechanical calculation (**SET MECH ON**) or suppress mechanical calculation (**SET MECH OFF**). The ON option is the default when starting FLAC.
Mono	sets graphics mode to monochrome, high resolution
Output	p sends plotted output to the device connected to port p where p can be COM1, COM2, LPT1 or any other port. Alternatively, p can be the name of a disk file. In this case, the plot file (HPGL or PostScript) will be stored in p. See, also, the **SET BAUD** command for setting output device baud rate.

SEt — 14

Pal	0	palette number 0. Line color choices are green, red and yellow-brown in CGA mode.
	1	palette number 1. Line color choices are cyan, magenta and white in CGA mode. Default = palette number 1. Colors for the EGA mode are given with the **PLOT** command.
PEA		will set HP pen plotter speed to fast.
Plot		name sets plotting device to HP pen plotter, Post-Script-compatible laser printer, or other printer, where name = HP, POST or PRINTER.

SEt —15

PLTF val PLTC val PLTT val	friction angle, cohesion and tensile strength parameters used for plots of Mohr-Coulomb safety factor.
PSL	will set HP pen plotter speed to slow.
Small	small strain (coordinates are not updated; this is the default)
STep n	sets the limit on number of steps for the SOLVE command.
UCS val	val is the unconfined compressive strength of the rock. It is used in plotting of contours of factors of safety for the Hoek-Brown model.

SEt ---- 16

VGA sets VGA graphics mode.

Some other quantities specific to groundwater, thermal and creep applications can also be set. (See the appropriate appendix for details.)

SOlve <keyword = value ...>

This command allows automatic detection of the steady-state solution of mechanical problems. Calculation is performed until the limiting condition for timestep, run time, or out-of-balance force is reached.

These limits are changed by the optional keywords.

Clock	run time (in minutes)
Step	timestep
Force	out-of-balance force

Defaults for these keywords are:

S = 500 steps
T = 5 minutes
F = 100

Note that F depends on the system of units adopted. Old limits will apply if not specified explicitly, but these limits are remembered when set or on restart. However, they are reset to defaults when a **NEW** command is given. A time limit of greater than 1440 minutes (24 hours) will not be accepted; for greater run time (e.g., over a weekend), several **SOLVE** commands can be given in sequence. If the "escape" key is touched during execution, FLAC will return control to the user after the current step is completed.

SOlve — 2

The **SOLVE** command can also be used to execute groundwater flow and thermal steps, if appropriate. (See the appropriate appendix for a discussion of this.)

STEp n

Execute n timesteps. If the "escape" key is touched during execution, FLAC will return control to the user after the current step is completed. The synonym **CYCLE** may also be used.

STop FLAC stops, and the session ends.

STRuct <keyword> . . .

The **STRUCT** command is used to define the geometry, properties, and end conditions for structural elements. The **STRUCT** command describes the element type and geometry, links the element to the FLAC grid, and associates a property type number with the element. The keywords used to define the structural element are as follows.

(1) Element Type Keyword

There are two element types:

Beam	A beam element is defined by the geometry and property keywords and values which follow the BEAM keyword.
Cable	A cable element is defined by the geometry and property keywords and values which follow the CABLE keyword.

Each beam or cable is defined by either beginning or ending coordinates, nodes, or main grid linkages. The beam or cable may be divided into a number of segments; each segment is connected by a node at each end. The

STRuct — 2

beam element node is described by three degrees of freedom: an x- and y-translation and a rotation. The cable element permits only x,y translations. Finally, properties are assigned through a property type number. Figure 4-4 illustrates the conditions associated with the beam element.

STRuct — 3

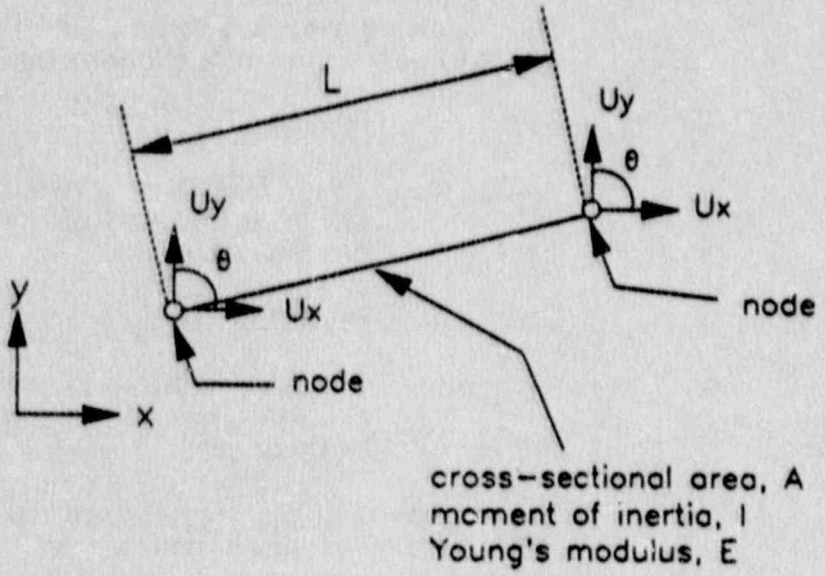


Fig. 4-4 Beam Element Geometry (3 degrees of freedom at each end node)

STRuct —4

(2) Element Geometry and Nodal Linkage

The following two keywords are required to define the element geometry.

Begin keyword

The BEGIN keyword identifies one end of the beam or cable.

End keyword

The END keyword identifies the other end of the beam or cable.

The endpoints of the element are defined by one of three possible keywords and/or values which immediately follow the BEGIN and END keywords:

Grid i,j

Here, a linkage is established between the beginning or ending element node and the soil or rock mass at gridpoint i,j

STRuct — 5

Node	n
	As the structure is created, FLAC will automatically give structural nodes a sequential number from 1 to the number of nodes in the order in which they were created. The beginning or ending of the present element may be coupled to node n by issuing this command.
x,y	An element node may be given a spatial coordinate simply by giving an x,y pair after the BEGIN or END keyword. It is possible to define elements at any orientation in space with this command. If x,y are very close to an existing node or gridpoint, FLAC will force this new node to conform to it.

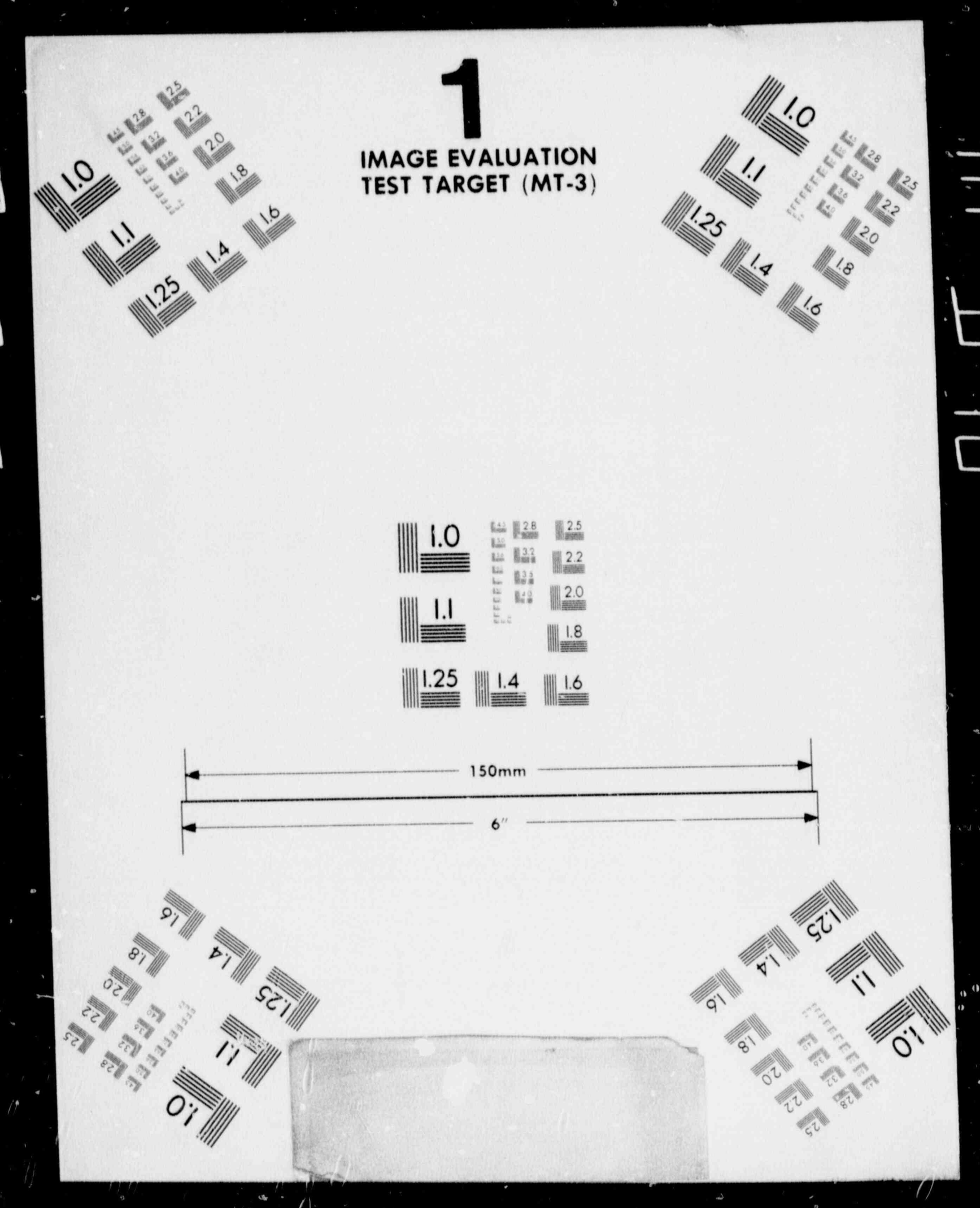
STRuct — 6

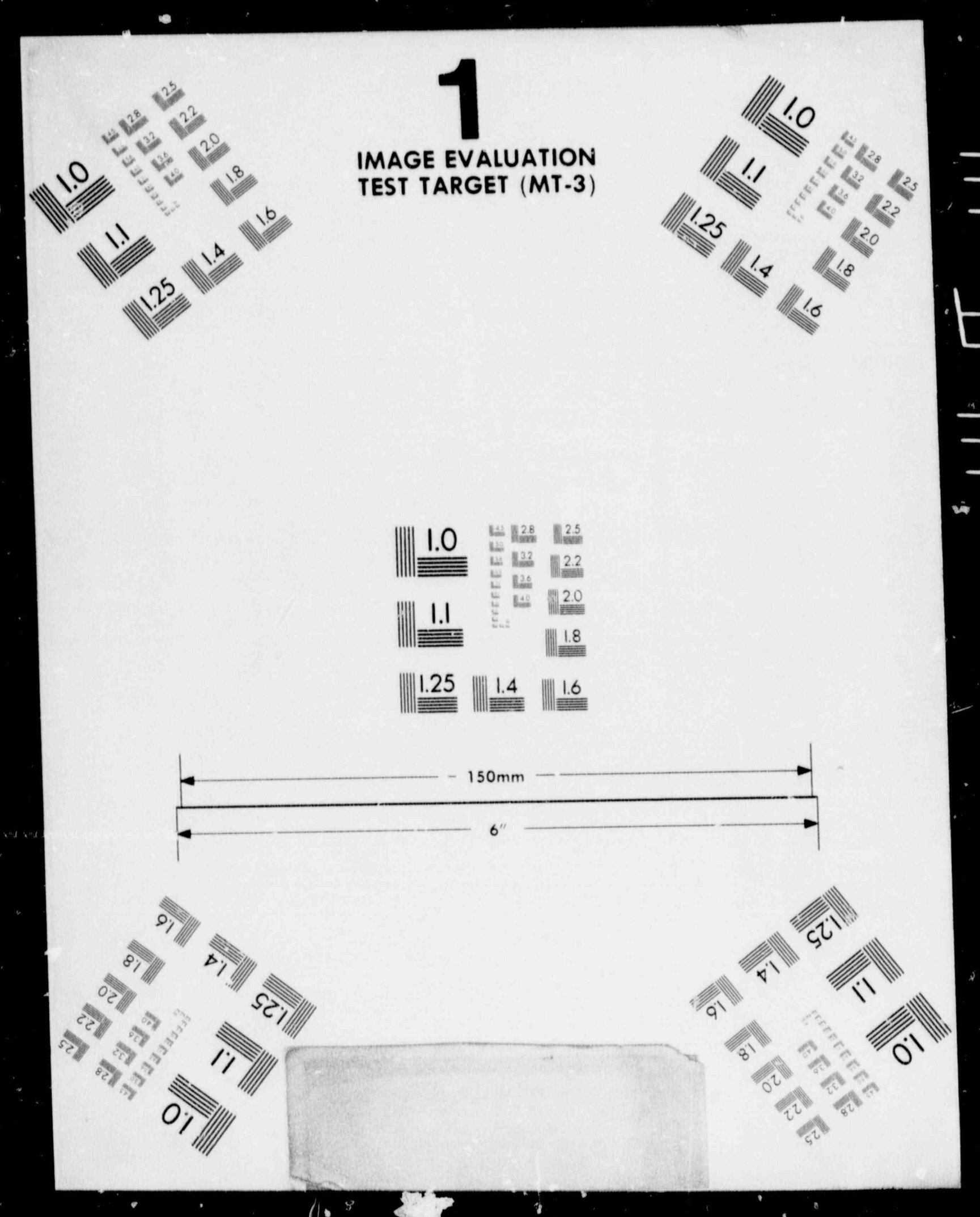
(3) Element Property Type Keyword

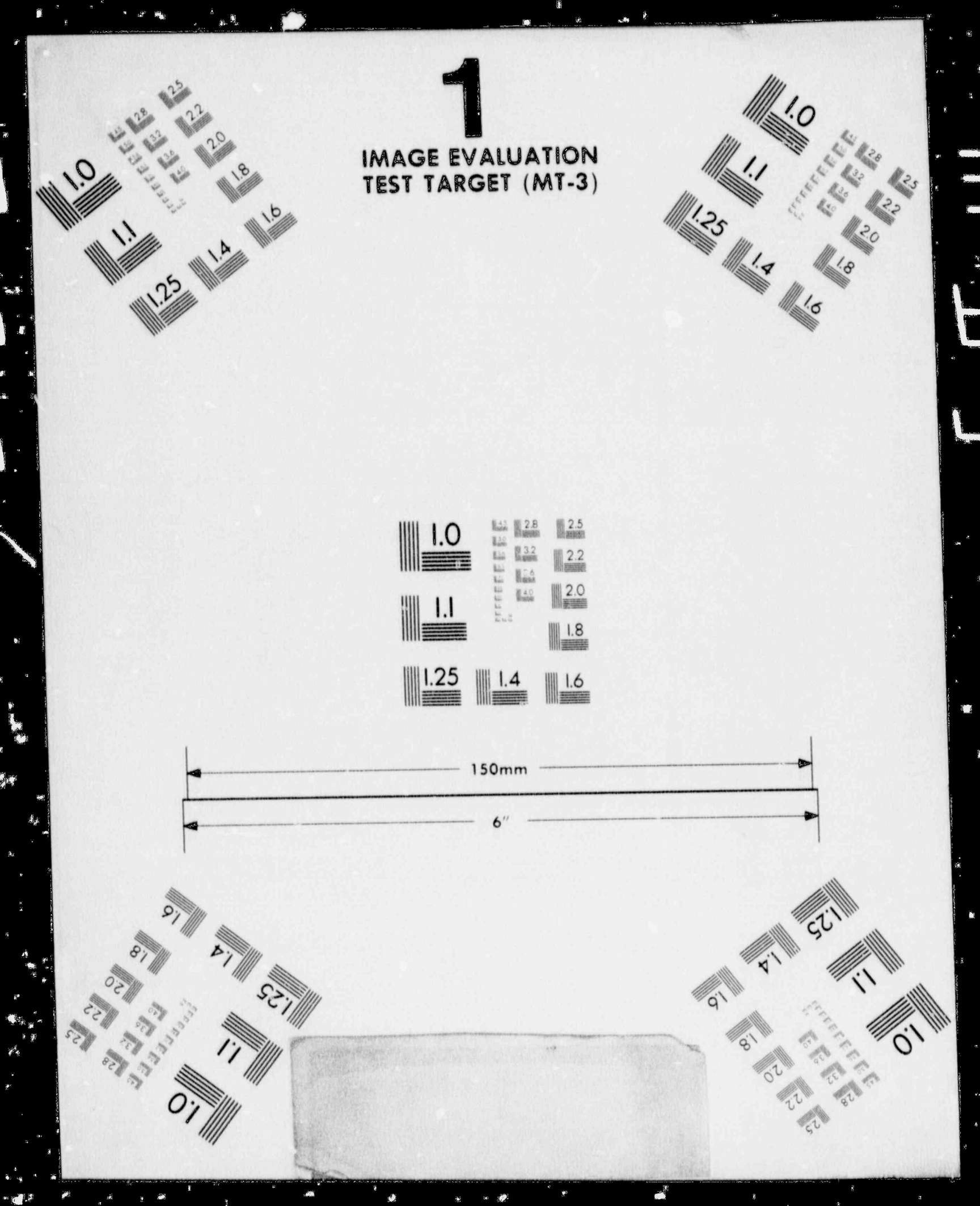
Each beam or cable may be assigned properties. This is done by associating a property number, n, with each beam or cable. The form of the keyword is:

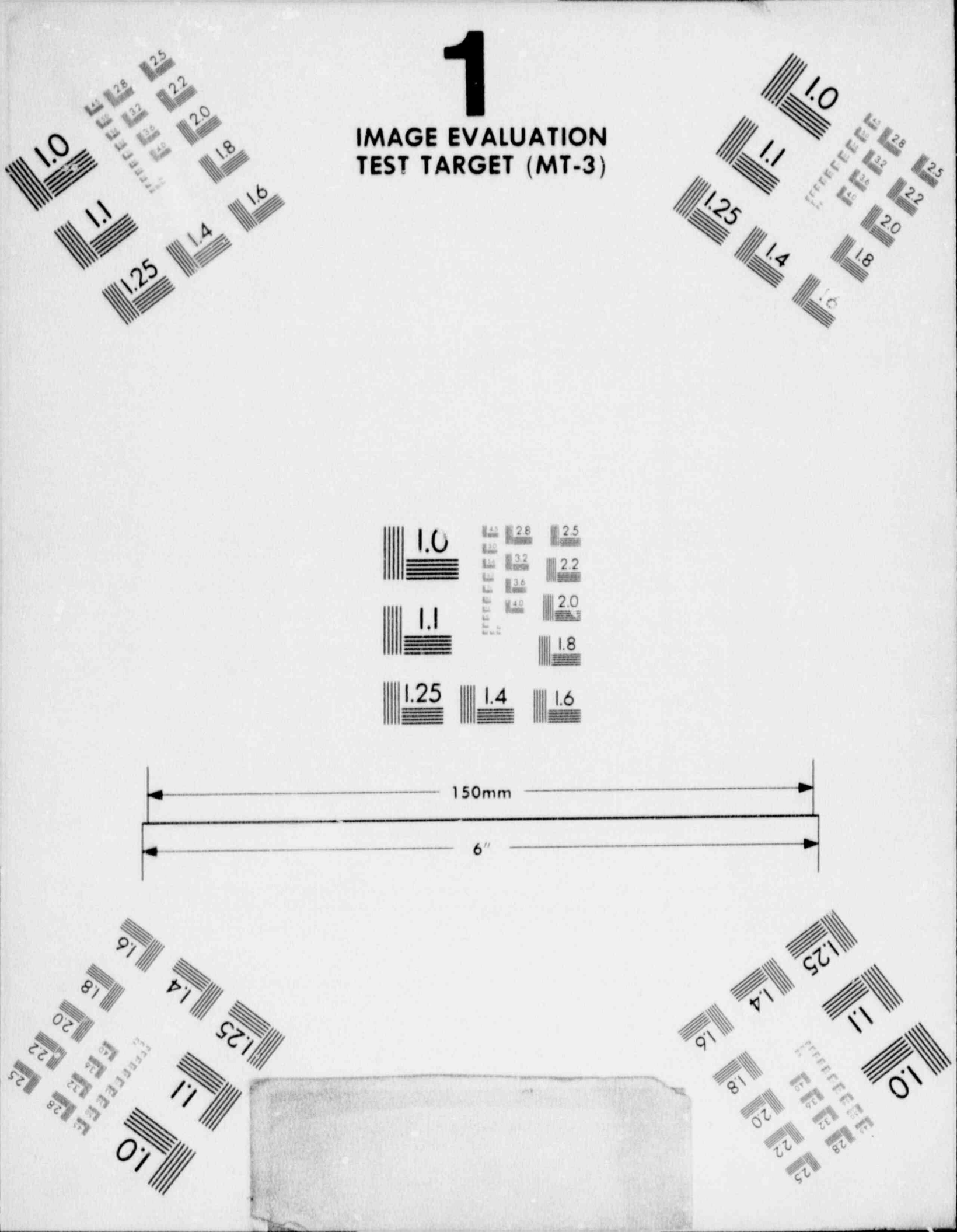
Prop n

Property number n is assigned to the element type. The properties are associated with the property number by using the command **struct prop=n keyword** described below.


1
IMAGE EVALUATION
TEST TARGET (MT-3)
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.5
4.0
150mm
6"


1
IMAGE EVALUATION
TEST TARGET (MT-3)
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
150mm
6"


1
IMAGE EVALUATION
TEST TARGET (MT-3)
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
4.0
150mm
6"


1
IMAGE EVALUATION
TEST TARGET (MT-3)
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
150mm
6"

STRuct — 7

(4) Element Segments Keyword

Each beam or cable may be divided into a number of segments by using the following keyword:

Seg k

The element is divided into k segments. For example, if the beginning coordinate is (0,0), the ending coordinate is (10,0) and k=10, then the element will be divided into 10 segments of length 1.0.

(5) Cable Pre-Tensioning Keyword

A cable bolt may be pre-tensioned to a load, t, by the keyword

Tension t

Two supplemental **STRuct** commands are provided to define element conditions further.

STRuct — 8

STRuct Node=n keyword

Each structural node may be prescribed various boundary or geometry conditions which are defined by the following keywords.

Fix <X> <Y> <R>

This option allows node n to have fixed x- and/or y-displacements or rotations (i.e., a beam end may be locked in place or allowed to rotate).

FRee <X> <Y> <R>

This is the opposite of **Fix**.

Pin This establishes a pin connection at node n (i.e., frees moments).

Load fx,fy,m

This allows the user to apply x- and/or y-direction forces or moments to node n.

STRuct — 9

x,y	These are the x- and y-coordinates of node n. This is used in establishing a new node which can then be included in the structural element. This permits the user to define node numbers prior to generation of the element.

STRuct — 10

STRuct Prop=n keyword = value . . .

This command associates property values to a particular property number n. Figures 4-5 and 4-6 illustrate the relation of grout and cable material properties, respectively. The following property keywords apply:

E = value

Young's modulus of beam or cable

I = value

moment of inertia of beam (second moment of area)

Area = value

cross-sectional area of beam or cable

Kbond = value

bond stiffness of grout [force/unit cable length/ displacement]

STRuct — 11

SBond = value

bond strength of grout [force/unit cable length]

Yield = value

yield strength (force) of cable

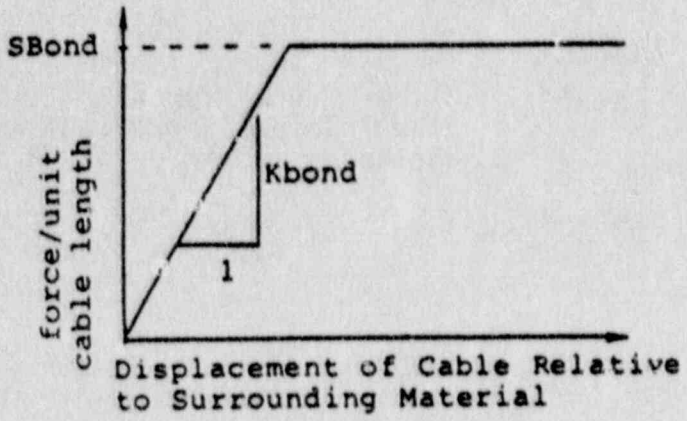


Fig 4-5 Grout Material Properties

STRuct — 12

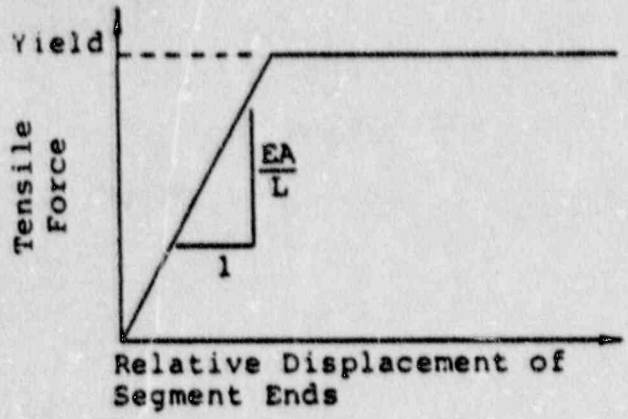


Fig. 4-6 Cable Material Properties (Note: Cable elements consider tensile forces only)

SYS

This command allows the user to access the DOS system for performing a number of standard system commands. A sys> will appear. The system commands are:

RENAME file1 file2;

COPY file1 file2;

DELETE file;

DIR <file-spec>; and

TYPE file.

The file-specification for **DIR** may include the "wild characters" ? or *, but these must not be used with the other commands. Note that **COPY** will overwrite file2 if such a filename already exists. A directory listing may be halted by hitting any key.

The user may return to FLAC by hitting the return key. An alternative form of the **SYS** command is to issue the entire command; for example,

SYS RENAME file1 file2

SYS —2

CAUTION: Do not use these commands with files FLAC currently has open, such as the file to which plots are being sent. If you wish to copy such a file, first close it (e.g., by SET **OUT=newname**, where "newname" is a different name). If these commands attempt to access open files, the system may hang up, and the files may be lost.

Table n <insert> x1 y1 <x2 y2> <x3 y3>. . .

This command sets up a table of x- and y-values in memory for use by FLAC. Tables are used in defining the water table for pore pressure calculation (see **WATER** command), the friction, cohesion and dilation tables for the strain-softening model (see **PROP** command), and a series of coordinates from which boundaries may be fit within the model grid (see the **GEN** command). Apart from the amount of memory remaining, there is no limit to the number of tables (see **PRINT MEM**). Any number of x,y pairs may be given; the number n identifies the table and may be any integer except zero. Items are added to the end of the current table unless the **INSERT** keyword is given. In that case, each item is inserted between the two existing items which bracket the given x-value. If an x-value for a given item is identical to that of an existing item, the existing item's y-value is updated (in **INSERT** mode). The command **PRINT TABLES** can be used to verify the contents of a table. Individual tables can also be plotted using the **PLOT TABLE** n command.

Title The next input line is taken as a title printed on subsequent pen and screen plots and recorded on restart files. If the **TITLE** command is given in interactive mode, the old title appears. If the "escape" key is hit, this title is retained; otherwise, a new title may be entered.

Unmark <range>

Existing marks on any gridpoint within the range are removed (see **MARK** and **GEN** comands).

WAter <Density = value> <Table = n> <Bulk=value>

A pore pressure is deposited in each zone according to the depth of the zone below a water line given by the specified table. The pressure is also proportional to the density of water (as given above) and the vertical acceleration of gravity (given on the **SET GRAVITY** command). Zones above the water line have zero pore pressure.

When pore pressures are deposited in the grid with the **WATER** command, direct stresses (sxx, syy) are increased by the same amount, since they all correspond to total stresses. The pore pressure distribution may be plotted and printed; pressures are taken as positive. The water line may also be displayed (using the **PLOT WATER** command).

During calculation, FLAC uses effective stresses (i.e., total stresses minus pore pressure) in consulting constitutive models. Pore pressures are not affected by zone volume changes, nor is there any flow of water unless the **CONFIG GW** command has been given. Total stresses are displayed on plots and printouts.

WAter — 2

If the command **WATER TABLE = 0** is given, all pore pressures are removed from the grid. Similarly, the pressure distribution may be changed during a run by switching to a new table or by altering the contents of the current table.

The bulk modulus of water may also be set with this command.

Window <xlo xhi ylo yhi>
<AUTO>

This command creates an imaginary window on the screen or pen plotter for the purposes of plotting. The region of space xlo to xhi, ylo to yhi is mapped onto the square screen area. Hence, if the window region is not square, a distorted picture will be drawn in which the x and y scales are different.

If the window is not defined prior to plotting, one is selected which encompasses the entire grid. The window will remain as set until changed. If automatic windowing is desired, use the AUTO keyword. If the window is less than the grid dimensions, the screen image will be clipped at the window boundaries. The user may use this feature to obtain enlarged views of detail at points of interest.

5.0 PROBLEM SOLVING WITH FLAC

5.1 Introduction

To run a simulation with FLAC, the problem geometry, boundary conditions, and material model must first be determined. These procedures are similar to those of nearly all stress analysis codes. The region to be modeled must be subdivided into a number of quadrilateral elements, or zones, with each zone defined by its four corner nodes or gridpoints. Internally, FLAC further divides each quadrilateral into four triangular elements for the subsequent calculations.

Once the mesh has been defined, boundary and initial conditions must be provided. The conditions can consist of:

(1) fixing gridpoint displacements or velocities in the x- and/or y-directions;

(2) applying pressure across any face or line of gridpoints;

(3) applying x- and/or y-direction forces at any gridpoint;

(4) initial stresses in the body; and

(5) gravity.

The actual solution of the problem is somewhat different for an explicit model such as FLAC than for many of the commonplace implicit codes. Explicit codes are often termed "time-marching" in that the basic equations of motion are solved at

successive timesteps until equilibrium is reached. (This was discussed in Chapter 3.0.) In practical terms, this means that, for the user, the problem is solved in a physically-meaningful manner.

For example, consider a tunnel excavated in a soil near the surface. In this case, the forces due to gravity are important. The geometry of the mesh may be set initially to a size large enough to avoid boundary effects [Fig. 5-1]. The side and lower boundaries are fixed on rollers and gravity is applied to the material. The model is then "stepped" through a number of timesteps until the velocity or out-of-balance force at the gridpoints approaches zero—i.e., until the vibrations have damped to an acceptable level. FLAC allows two basic methods for time-stepping to equilibrium at each model stage. The user may give a specific number of timesteps which are desired, the limit on maximum out-of-balance force within the grid or the real time (in minutes) which is to be allowed for solution. Alternatively, he or she may allow the program to solve the problem to equilibrium without user interference. The advantage in the first approach results from the explicit nature of the solution method. Explicit programs were originally developed to allow solution of highly nonlinear problems such as structural collapse in which equilibrium may never be obtained. In this case, it is advantageous to be able to examine the collapse process as it occurs. To do this, the solution process is cycled a number of steps at a time (depending on the type of problem), a "save" file is created, plots made, and numerical output examined. The save file is restored and cycling continued in this manner to the satisfaction of the user (i.e., either to equilibrium or to some failure state).

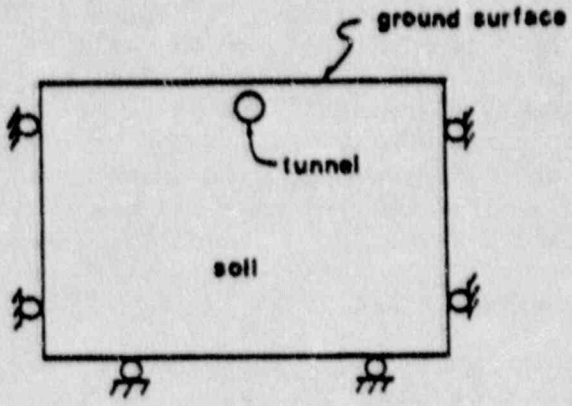


Fig. 5-1 Near-Surface Tunnel Problem

Consider the near surface tunnel problem illustrated in Fig. 5-1. In this example, once equilibrium has occurred in the grid without the excavation, the gravitational stresses in the body will have been established and will be at the in-situ stress state. At this point, the excavation is made by changing the material model of these zones to the "null" model. The calculation process then proceeds by again stepping to equilibrium. As before, the user can simply allow the program to solve for the equilibrium state or can step manually and examine the state of the body at intermediate steps.

The problem-solving process with FLAC therefore involves the logic illustrated in Fig. 5-2.

This procedure is convenient because it physically represents the processes which occur in the natural environment. Modeling of a few simple problems such as Prandtl's wedge, a circular tunnel at great depth (Kirsch's solution), and soil slope stability under gravitational loads should be run to familiarize the user with the solution process.

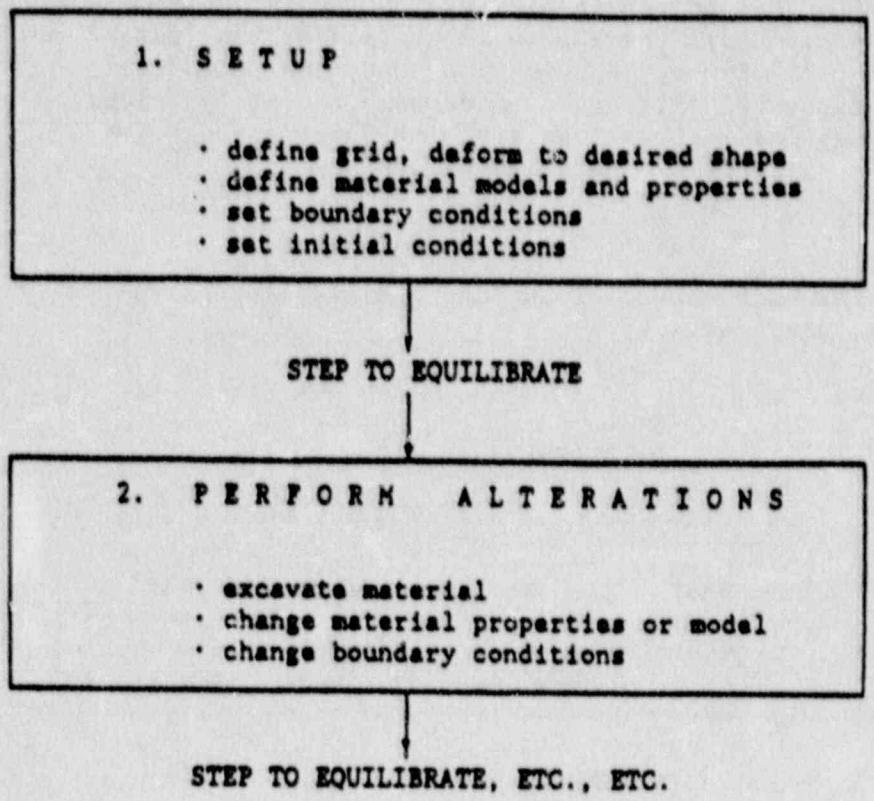


Fig. 5-2 General Solution Procedure for an Explicit Numerical Model

5.2 Running FLAC

FLAC can be run in either an interactive (commands entered via the keyboard) or file-driven (batch) mode. First, the user must copy the code from floppy to hard disk and de-compress it by following the instructions given in Chapter 1.0. The program is then started by typing

C> **FLAC**

The machine will load FLAC and display the following message:

FLAC (Version 2.00)

```
----------------------------------------------
|                                            |
|      Fast Lagrangian Analysis of Continua  |
|--------------------------------------------|
|(C)Copyright Peter Cundall (1987)           |
|             Itasca Consulting Group, Inc.(1987)|
----------------------------------------------
```

Strike return to continue

If a FLAC.INI file exists, the program will read and execute the commands when the return key is hit. When FLAC is ready to receive commands, the following prompt is displayed:*

flac>

If you wish to run the code interactively, just begin typing in commands from the list presented earlier and discussed further in the following sections. FLAC will execute each command as the return key is pressed. If an error arises, a warning to the user will be written to the screen.

The code can also be run from a previously-prepared data file which contains a set of commands just as they would be entered in the interactive mode. The data file can be created using any DOS editor and stored using the file extension **.dat**. To recall and execute the data file, the **CALL** command is used:

flac> **call file.dat**

where file is the name you have given to the data file. You will see the data entries scroll up on the screen as FLAC reads each line. Finally, the program can be restarted from a previously-saved state using the restore option. Typical data sets for all three of these input options are given in the following section.

*Note: In the following discussions, the flac prompt (flac>) is shown before user input commands.

5.2.1 Mesh Generation and Material Properties Definition — The first input commands which must be given generate the finite difference mesh. The three primary commands which control this function are:

flac> **grid icol,jrow**

flac> **gen***

flac> **initial x or y <range>**

The **GRID** command generates a mesh icol zones wide by jrow zones high. The i-axis is parallel to the problem x-axis and the j-axis is parallel to the problem y-axis (Fig. 4-1).

The **GEN** command is then used to provide coordinates for the mesh and to deform the mesh to any desired shape. Specific features of the **GEN** command include "grading" the mesh to represent far boundaries.

This command is also used to create boundaries of various shapes within (or on the exterior of) the mesh. Pre-programmed shapes include circles, arcs and lines. The line function can further be used to create arbitrarily-shaped openings. The following are examples of the generation of shapes using the **GEN** command.

---

*See Section 4.0 for the various forms of the **GEN** command.

Example 1

In its simplest form, the **GEN** command can supply new coordinates to a grid. The commands

```
flac> grid 10,10
flac> m e
```

will create a square grid of 10 zones by 10 zones (11 gridpoints by 11 gridpoints) and will assign an elastic model to each. If the coordinates of the grid are printed at this stage by issuing the command

```
flac> pr x  y
```

you will note that both run from 0.0 to 10.0 —i.e., FLAC assigns a square grid with 1 unit spacing between gridpoints. If the actual coordinates of the grid are to run from 0.0 to 500.0 in the x-direction and 0.0 to 1000.0 in the y-direction, the **GEN** command is used as follows:

```
flac> gen 0.0,0.0  0.0,1000.0  500.0,1000.0
          500.0,0.0  i=1,11  j=1,11
```

Note that the portion of the mesh defined by i=1,11, j=1,11 starts at the lower left-hand corner of the grid and works around its outer corners in a clockwise fashion. All gridpoints interior to these corner points will have their coordinates re-assigned based on the corner point coordinates. Now, print out the coordinates again to see that the coordinates have, indeed, been changed. Note that

just a portion of the grid can be given new coordinates. The portion of the grid is defined by the i,j range. In the previous example, if i=5,11, j=5,11 were specified as the range, only the upper right-hand corner of the grid would be affected. The range must be specified in a clockwise fashion.

Example 2

The **GEN** command can be used to create distortions in the grid. For example, try the following commands:

```
flac> new
flac> grid 20,20
flac> m e
flac> gen 0,5 0,20 20,20 5,5 i=1,11
flac> gen same same 20,0 5,0 i=11,21
flac> plot grid
```

In this example, only a portion of the grid is distorted with each **GEN** command. The first command creates a distorted quadrilateral from half of the grid while the second command "wraps" the remainder of the grid around to form a rectangular opening. Note that successive **GEN** commands are additive—i.e., once changed, the coordinates of the grid remain at the new coordinates until changed again by using the **GEN** or **INI** commands.

Example 3

The **GEN** command can be used to grade a mesh to represent far boundaries. For example, in many cases, an excavation is to be created at a great depth in a rock mass. Detailed information on the stresses and displacements is to be determined around the excavation, where the disturbance is large, but little detail is necessary at greater distances. In the following example, the lower left-hand portion of the grid is left tightly discretized and the boundaries are graded outward in the x- and y-direction. Try issuing the following commands:

```
flac> new
flac> grid 20,20
flac> gen e
flac>
flac> plo grid
```

The **GEN** command forces the gridlines to expand to 100.0 units at a rate 1.25 times the previous grid spacing in the x- and y-directions. Note that if the ratio is between 0 and 1, the grid dimensions will decrease with increasing coordinate value. For example, issue the following commands:

```
flac> new
flac> gr 10,10
flac> m e
flac> gen -100,0  -100,100  0,100  0,0  rat .80,1.25
flac> plot grid
```

You will see a grid graded in the -x and +y directions.

Example 4

Excavations often need to be created in the grid. It is very tedious to create complex excavation shapes, especially circular arcs, by simply moving individual gridpoints. Two examples are given here for the creation of excavation shapes using the **GEN** command.

First, a circular excavation is created. Try the following commands:

```
flac> grid 20,20
flac> m e
flac> gen circle  10,10  5
flac> plot grid
```

This command automatically creates a circular opening outline within the grid, centered at (x=10, y=10) and with a radius of 5.0. Note that the remainder of the mesh remains square —i.e., element corners are at 90 degrees.

To cause the mesh to conform better to the new opening, type

```
flac> gen adjust
flac> plot grid
```

Successive **GEN ADJUST** commands will warp the grid to increasingly greater levels.

When creating internal shapes within the grid using the **CIRCLE**, **ARC**, or **LINE** commands FLAC distinguishes between the various regions of the grid created by marking closed paths. In the previous example, the **GEN CIRCLE** command creates two regions within the grid created by the boundary of the circle: the region inside the boundary and that outside. If you wish to see where the boundaries of the grid are, type

```
flac> pr mark
```

Those zones which have been adjusted by FLAC to conform to boundaries are signified by an "M" in the printout.

One note of caution—two regions can only be formed if they are separated by closed contours. In other words, a line segment which begins and ends within the grid and does not form a closed boundary will subsequently result in only one region. For the previous example, the REGION keyword can be used to assign models, properties, or initial values to either region. For example, to null out the interior of the tunnel from the previous example requires only the command

```
flac> mod null  region=10,10
flac> plot grid
```

You will note that the tunnel zones have been excavated. They can now be replaced by typing

```
flac> m e  region=10,10
```

Note that the excavation can be replaced by any model and with properties consistent with the model. See the **MODEL** and **PROP** commands for greater detail.

Example 5

INITIAL can be used to move a point or a number of points from the present to a new location. The following commands create a grid and distort it using the **INITIAL** command:

```
flac> grid  5,5
flac> model elastic
flac> gen 0,0  0,10  10,10  10,0 (assigns coordinates to gridpoint from 0 to 10 in the x- and y-directions)
flac> ini x=-12  i=1  j=6 (moves upper left-hand corner horizontally by -2 units)
flac> ini x=12  i=6 (moves right-hand boundary gridpoints to the right by 2 units; note that, since the j range is not given, the entire range is assumed)
```

Note that the **INITIAL** command can be used to move any gridpoint to any position. Of course, elements cannot overlap. If this happens, a warning message referring to "BAD GEOMETRY" will be given

and FLAC will not continue execution until the errors in grid construction are rectified. A practical limit on the aspect ratio of zones should be kept to about 1:10 or less for reasonable solution accuracy.

Once the mesh is defined, one or more material models and the required properties must be assigned to the zones within the body. This is done by using the **MODEL** and **PROPERTY** commands described in Section 3.0.

5.2.2 Applying the Boundary Conditions — After the mesh has been generated boundary and/or initial conditions are applied. The possible boundary conditions are shown in Table 5-1.

Table 5-1

BOUNDARY AND INITIAL CONDITION COMMAND SUMMARY

COMMAND		EFFECT
Initial		
	SXX	initialize sxx stress for a zone(s)
	SYY	initialize syy stress for a zone(s)
	SXY	initialize sxy stress for a zone(s)
	XDIS	initialize x-displacement for a zone(s)
	YDIS	initialize y-displacement for a zone(s)
	XVEL	initialize x-velocity for a zone(s)
	YVEL	initialize y-velocity for a zone(s)
FIX	X	fix velocity (or displacements) in x-direction at a gridpoint
	Y	fix velocity (or displacements) in y-direction at a gridpoint
	X Y	fix velocity (or displacements) in x- and y-directions at a gridpoint
Free	X	opposite of fix command
	Y	opposite of fix command
	X Y	opposite of fix command
Apply	Pressure	pressure over a row or column of gridpoints
	XFORCE	x-force applied over rows or columns of gridpoints
	YFORCE	y-force applied over rows or columns of gridpoints

For example, consider the development of a grid subjected to gravity load. Within this grid, a near-surface circular tunnel will be mined. The following command sequence may be typed interactively to examine the effects of the GEN command:

```
flac> grid 10,10
flac> model elastic
flac> gen circle 5,5,2 (deform grid to accommodate
          tunnel prior to initializing the gravi-
          tational stresses)
flac> plot grid
flac> gen adjust (adjust grid)
flac> plot grid
flac> prop s=.3e8  b=1e8  d=1600 (properties)
flac> set grav=9.81
flac> fix x  i=1   | (boundary condition rollers on
flac> fix x  i=11  | sides and bottom to let soil
flac> fix y  j=1   | settle uniformly)
flac> step 200
flac> * stresses developed in body; now, excavate
                tunnel and see reaction
flac> pr mark (shows the boundaries of regions
          within the grid)
flac> mod null region=5,5  (null out region inside
          tunnel)
flac> plot grid
flac> step 300
flac> plo str  bou
. . . continue simulation as desired
```

The above problem illustrates an initial stress problem—i.e., the stresses exist in the body prior to excavation. Because the grid cannot be altered after timestepping starts, it must be deformed to fit the boundaries of the grid before

the initial stresses are equilibrated. This type of problem is in contrast to one in which the tunnel exists before the body is stressed. The ultimate stress state will be the same for the case of an unsupported excavation; however, the displacements at the tunnel will be different. This is analogous to cutting a hole in a pre-stressed plate compared to cutting a hole in a plate and subsequently stressing the plate. The user must decide which type of problem is appropriate under the circumstances.

5.2.3 Applying Load/Changing Conditions — FLAC allows model conditions to be changed at any point in the solution process. These changes may be of the form:

- excavation of material
- addition or deletion of gridpoint loads or pressures
- change of material model or properties for any zone
- fix or free displacements or velocities for any gridpoint

To perform these operations in the interactive mode, simply enter the new commands, one at a time, at the screen prompt.

As an example of the ability to change conditions during a run, consider a simple example of cut-and-fill mining, where excavations are created and backfilled sequentially. Try running the following simplified problem:

```
flac> grid 10,10
flac> m e  *elastic rock mass
flac> prop s=5.7e9  b=11.1e9  d=2000
flac> fix x  y  i=1   |
flac> fix x  y  j=1   |
flac> fix x  y  i=11  | fix all boundaries
flac> fix x  y  j=11  |
flac> ini sxx=-30e6  syy=-20e6 (in-situ stresses)
flac> his nstep=1
flac> his xdis  i=5 j=5 (displacement history at
        stope wall)
flac> solve
flac> * now excavate
flac> mod null  i=3,4  j=3,6 (excavation # 1)
flac> plot grid
flac> step 200 (step to equilibrium)
flac> plo his 1 (check equilibrium)
flac> plot grid str (plot stresses, grid)
flac> mod mohr  i=3,4  j=3,6 (fill excavation with
        sandfill)
flac> prop s=.3e8  b=1e8  fric=30  i=3,4  j=3,6
        (sandfill properties)
flac> mod null i=3,4  j=7 (excavation # 2)
flac> ini xd=0  yd=0 (initialize displacements so
        that the changes due to excavation can
        be seen)
flac> step 200 (step to near equilibrium again)
flac> plo his 1
flac> plot grid str (plot stresses and grid)
flac> plot coh (plot different material types)
```

In this example, the boundaries are far too close for an accurate solution, but it illustrates FLAC's ability to change the model conditions during a run. Note that, if the user plots the grid after replacing the null zones with another material (in this case, a sandfill), the elements removed will be replaced. This points out that, even though the zones are given null properties and are not plotted or printed, they still exist and can be altered at any time. A contour plot of the material properties may also be plotted by typing **plot** followed by the name of the desired property.

The use of the history plot shown here enables the user to judge when the problem is sufficiently close to equilibrium. In this case, the x-displacement of a gridpoint at the excavation wall is used as an indicator of the movement toward equilibrium. When there is little change in value (displacement, stress, etc.) between successive timesteps, the model is more-or-less at equilibrium.

5.2.4 Printing and Plotting of Data — As previously discussed, a screen print of pertinent data can be obtained by simply using the print command:

```
flac> print  keyword
```

The matrix of values printed is in a logical format analogous to the grid itself. The keywords are self-explanatory, with two exceptions:

1. As previously discussed, FLAC subdivides the user-defined rectangular zones into four triangular sub-zones which are designated a,b,c and d. The user may request a print of the stresses within each subzone by using the commands:

   flac> **print asxx, ... etc.**

2. FLAC keeps a record of the yield history of each zone. It is often useful to know which zones are currently yielding, those which have yielded in the past but are currently elastic, and those which are now and have been elastic. The **PRINT STATE** command can be used to provide the current plastic state of each zone. The numbers printed in the matrix table have the following meaning:

   0 elastic
   1 at yield
   2 elastic but previously at yield
   3 has surpassed tension cut-off in uniaxial tension
   4 at yield, also surpassed tension cut-off
   5 has surpassed tension cut-off in general tension
   6 ubiquitous joints currently at yield
   7 ubiquitous joints have yielded in the past but are currently in the elastic state

The user may obtain printed copies of any text typed to the screen by using the **SET LOG ON** and **SET LOG OFF** facility. In a FLAC data set (either from a batch file or an interactive file), the command

flac> set log=on

will open a hard disk file (FLAC.LOG) which will then record all the commands which follow and all FLAC responses. To turn off the logging facility, type

flac> set log=off

This file can then be examined or copied to the printer for permanent record after exiting FLAC. Simply use the DOS copy or print commands.

Plots of the grid geometry and output variables can be made on the PC display or on a Hewlett-Packard or compatible plotter. The current plot types available to the user are listed below.

1. **GRID** plots the finite difference grid. As timestepping continues, the **GRID** command will plot the deformed shape (in large strain mode) of the finite difference grid . This is particularly important with large strain logic since, often, failure modes become evident from the deformed plots.

2. **BOUNDARY** plots the boundaries of the grid and excavations. It is often advantageous to overlay the boundary plot on the other variable plots such as vector or contour plots.

3. **DISPLACEMENT** and **VELOCITY** draws vector plots of displacement or velocity at each gridpoint.

4. **SXX**, **SYY** and **SXY** plots the xx, yy, or xy stresses.

5. **XDISP** and **YDISP** plot the x-displacement and y-displacement contours.

6. **XVEL** and **YVEL** plot the x-velocity and y-velocity.

7. **STRESS** plots the principal stress vectors.

8. **PP** plots the pore pressure contours.

9. **HISTORY** plots time histories of variables selected with the **HIS** command.

10. **STATE** plots the yield state of the zones.

11. **BEAM** plots structural elements.

12. **CABLE** plots cable bolt geometry.

13. **RF** plots reaction forces

The graphics screen must first be initialized by using the **SET** command. Version 2.00 requires an IBM (or compatible) CGA or EGA graphics card or the Hercules monochrome card. The **SET** command defines:

(1) the type of graphics screen* — low resolution color (CGA), high resolution color (EGA), or high-resolution monochrome. The default is

*Monochrome plotting requires the color graphics adaptor but will plot in the optional high resolution monochrome graphics mode on the color screen. A separate version of FLAC is available which works with the Hercules Graphics Board.

the low-resolution color screen. The choice of screen is set using either the **SET CGA** (for a low resolution color graphics adaptor), **SET MONO** (for monochrome), or **SET EGA** (for an enhanced graphics adaptor) commands.

(2) the background color — The background color is set using **set back** = iback, where iback is in the range 0≤iback≤7, as follows:

0 = black
1 = blue
2 = green
3 = cyan
4 = red
5 = magenta
6 = yellow-brown
7 = white

The default is green in monochrome mode because the background is the line color. If you desire, for example, red plots on a black screen in monochrome mode, **set back** = 4. The CGA and EGA boards require a black background (or **bac=0**). If you have either of these options, it is best to create a FLAC.INI file with **set bac** = 0.

(3) the palette selection — There are two possible palette selections (ipal=0 or 1) set using the **SET PAL** command. Each palette has three possible foreground colors. For ipal=0, the CGA color choices are green, red, and yellow-brown. For ipal=1, the CGA color choices are

cyan, magenta, and white. For the EGA, 14 color choices are possible. See the **PLOT** command (Section 4) for details on color choices. Foreground colors are defined by the color keyword in the **PLOT** command.

An example command set for a color, low resolution screen with black background, palette #0 is:

```
flac> set  pal = 0  back = 0  CGA
```

Once the graphics device has been initialized, the plot window may be set. The window should be square to eliminate plot distortion. The form of the command is:

```
flac> wind  xlow,xhigh,ylow,yhigh
```

If the **WINDOW** command is not given, a scale will be chosen automatically. Note that, by changing the window dimensions, plots may be enlarged for detailed examination of the grid. The window should always remain square—otherwise, the plots will be distorted. Plots may have a slight distortion, even with a square window. This is due to the varying vertical adjustment of each screen. To eliminate any distortion, adjust the vertical adjust knob on the back of the monitor or use the **SET ASPECT** command.

Plots can now be made to the screen by simply typing the command word **plot** followed by the keywords for the plots which are to be overlaid. An example plot command is:

```
flac> plot sxx  disp  boun
```

which will overlay the sxx stress contours, the displacement vectors, and the grid and excavation boundaries. A heading describing the plot is produced in the upper left-hand corner. Further enhancements to the plot can be obtained by using "sub-keywords" or switches. These switches can be used to:

(1) define the maximum vector length (max = value);

(2) define the contour interval (int = value);

(3) define line color; and

(4) remove headings.

For example, the above-noted command could be given as:

```
flac> plot sxx int=1E4 cyan disp max=.001 mag
      boun
```

Here, the sxx stress contours are plotted in cyan (assuming palette #1) with a contour interval of 1E04, the displacement vectors in magenta are scaled to a maximum vector length of .001, and the boundary of the grid and excavations are in the background color.

Contours may be labeled by using the **SCLIN** command prior to plotting.

Once a plot has been produced on the screen and is satisfactory, a pen copy of the plot can be produced on a Hewlett-Packard pen plotter. A plot is

obtained by typing the command **PEN** before the plot keywords. For example, the command

```
flac> plot pen  sxx  disp  boun
```

will dump the plot to the pen plotter.

The plot can also be sent to a graphics dot matrix printer. This is done by issuing the DOS command **GRAPHICS** prior to loading FLAC. When a plot is made on the screen, it can be dumped to the printer by using the print screen key. The plot generally will not have the same aspect ratio on the printer as on the screen. Some experimentation may be necessary, using a non-square window, to create a "corrected" distorted plot on the screen which will result in a non-distorted plot on the printer. Several commercial packages are available to create dot matrix plots from the CGA or EGA boards in monochrome or color.

5.3 Special Considerations

FLAC has several features of note which are reviewed here.

5.3.1 Large Strain — As the body undergoes strain, the normal assumption is that the magnitude of the displacement is small enough that the coordinates of the body remain unchanged. This is the case with the default small strain logic in FLAC. If, however, the strains are large, the coordinates of the gridpoints must be updated, and zone stresses corrected, as strain occurs. When dealing with plastic constitutive models, and pos-

sible structural collapse modes, the large strain logic is essential. This is the case with the large strain option in FLAC, which is invoked by using the command

```
flac> set large
```

If you wish to return to small strain mode, the command

```
flac> set small
```

is used. Gridpoint coordinates will then not be updated in future calculations.

5.3.2 Plane Stress — The default solution mode for FLAC is plane strain. However, the assumption of plane stress can be set by using the command

```
flac> set p-stress
```

It is not logical to switch from plane strain to plane stress in the middle of a run, so this command should only be used at the time of grid initialization.

5.3.3 Gravity — Gravitational forces can be induced in a body by setting gravity on

```
flac> set grav 9.8 0
```

The value of 9.8 is the acceleration due to gravity and 0° signifies that gravity acts vertically. (The default is 0 and is not necessary.) The sign convention for gravity is as follows:

+ = down, - = up.

5.3.4 Aspect — A square grid should appear as a square on your monitor screen. If it does not, there are two ways to correct for this.

The first method involves manual adjustment of the vertical size control or your PC monitor. If this does not resolve the problem, then draw a square grid on the screen (i.e., **grid 2,2**) and manually measure the horizontal and vertical dimensions with a scale. The aspect ratio is equal to the horizontal/vertical ratio. Set the aspect ratio in FLAC by typing

```
flac> set asp=a
```

5.3.5 Logging — A hard disk record of your FLAC session can be obtained by "logging on" and "logging off". This is done with the commands

```
flac> set log=on
```

and

```
flac> set log=off
```

to log on and log off, respectively. Anything which is printed to the screen between these commands will be written to a hard disk file called

FLAC.LOG. Remember that an existing FLAC.LOG file will be over-written when you start a new FLAC run. If you want to retain the log file, re-name it before starting FLAC.

5.4 Error Handling

FLAC has an error recovery logic which recognizes many of the common errors and gives the user a message without aborting the program when in interactive mode. Control is returned to the prompt level. A listing of the error messages is given in Appendix C.

5.5 Saving/Restoring Runs

FLAC allows the user to save a state and then restore it at any point in the simulation process. A save file is created by typing the command:

```
flac> save Z:file.sav
```

where "file" is the name you wish to give the save file and "Z" is the drive specification. To restore a saved file, simply type the command

```
flac> rest Z:file.sav
```

where, again, "file" is the name of the previously-saved state. It is always a good idea to make liberal use of save files for purposes such as parameter studies, plotting, and printing.

5.6 Suggestions and Advice

- FLAC uses constant-strain zones. If the strain gradient is high, you need many zones to represent the non-uniform strain distribution. Try running the same problem with more zones, to check. Constant-strain zones are used because, for plastic flow, it is better to use many low-order elements than a few high-order elements.

- Try to keep zoning as uniform as possible. Avoid long, thin zones or very distorted zones.

- For a new problem, always do a trial run with a few zones to get a quick feel for the response and possible difficulties. When you understand the trial results, increase the number of zones to obtain better accuracy.

- FLAC will take a longer time to converge if:

  (a) there are big contrasts in stiffnesses; or

  (b) there are big contrasts in zone sizes.

- A very stiff loading plate often can be replaced by a series of fixed gridpoints which are given constant velocity. (Recall that the **FIX** command fixes velocities, not displacements.)

- In order to determine a collapse load, it often is better to do it under "strain-controlled" conditions rather than "stress-controlled" conditions—i.e., apply a constant velocity and measure the reaction forces rather than applying forces and measuring displacements. A system that collapses becomes difficult to control as the applied load approaches the collapse load. (This is true of a real system as well as a model system.)

- Use symmetry conditions, whenever possible, to save computer memory and run time. For example, if a system is symmetrical about a vertical axis, you can represent the symmetry line as a vertical boundary with the gridpoints fixed in the x-direction (but free in the y-direction).

- Make frequent use of save files. For example, save intermediate states when doing parameter studies. If the run stops for any reason, you will have the intermediate states.

- Treat a FLAC model just like a physical model. Try to reproduce in a FLAC run the stages that actually would occur in nature. Keep in mind that there is no unique equilibrium state for an inelastic system. There may be many possible states that satisfy equilibrium; the one you get depends on the history.

- FLAC shows how a system behaves. Make frequent, simple tests to check that you are doing what you think you are doing. For example, if a loading condition and geometry is symmetrical, check that the response is symmetrical or, after making a loading change or other change, execute a few steps initially (e.g., 5) to verify that the initial response is of the correct sign and in the correct location. You might also do back-of-the-envelope estimates of the expected order of magnitude of stress or displacements and compare them to FLAC output.

- If you apply a violent shock to a system, you will get a violent response. If you do non-physically reasonable things to the system, you must expect strange results.

- Critically examine the output before proceeding with the simulations. If, for example, everything is ok except for large velocities in one corner zone, do not go on until you understand the reason. In this case, you might have left a "fixed" grid point free.

- FLAC does not give a "Factor of Safety" directly. If you need a factor of safety, it can be defined for any parameter that you consider important by taking the ratio of the actual value to the value which causes failure. For example,

$$F_w = \frac{\text{water level to cause collapse}}{\text{actual water level}}$$

$$F_\phi = \frac{\tan(\text{actual friction angle})}{\tan(\text{friction angle to cause failure})}$$

$$F_L = \frac{\text{load to cause failure}}{\text{design load}}$$

Note that the larger value is always divided by the smaller value (assuming that the system does not fail under the actual conditions).

- If you want FLAC to run overnight but still have the computer available the next morning, simply set it going with a large number of steps. You can use the "escape" key to return to the command level when you need the machine. It is safer to make long runs under the control of a CALLed file which contains frequent saves, in case of power failure.

- Use history plots to determine when a model is in equilibrium.

6.0 STRUCTURAL MODELING IN FLAC

Structures of arbitrary geometry and properties and their interaction with a soil or rock mass may be modeled with FLAC. Two types of structural elements are provided:

(1) beam elements (which can be joined together with one another and/or the soil or rock mass to model problems such as retaining walls, support struts in an open-cut excavation, and concrete or shotcrete linings for a tunnel or surface structure such as footings); and

(2) cable or rockbolt elements (which can be used to represent fully-grouted or point-anchor rockbolts, cablebolts, or tie-back anchors).

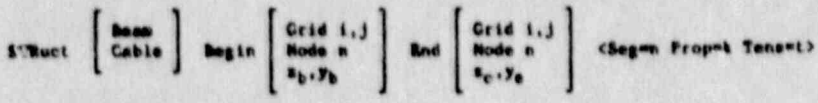
In all cases, the commands necessary to define the structure(s) are quite simple but invoke a very powerful and flexible structural logic. Because this structural logic is developed with the same finite difference logic as the rest of the code (as opposed to a matrix-structural approach), the structure may be subject to large displacements.

6.1 Command Structure

To model the interaction of structures with a soil or rock mass, the user must define the geometry of the support (in addition to the body), the properties of the support, and the type of linkage be-

tween the structural elements themselves and the support medium.

To perform these functions, three basic commands are provided. The command



defines the type of element, its beginning grid-point linkage, structural node or x,y coordinate, the ending reference point, the number of segments into which the element is to be divided, and the property type for the element. A cable bolt may be pre-tensioned to a load, t, by the keyword TENSION.

The command

STRuct Prop=k keyword . . .

defines the type of properties for the beam or cable bolt. The following property keywords may be used:

Area	cross-sectional area
E	Young's modulus
I	moment of inertia of beam (second moment of area)
Kbond	bond stiffness of grout
SBond	bond strength of grout
Yield	yield strength of cable

Finally, the command

STRuct Node=n keyword

defines the linkages at structural element nodes. The user has the choice of fixing or freeing displacements or rotations at nodes, applying x- or y-directed forces or moments or specifying a pin joint at structural nodes. By default, the nodes are considered to be free. The following keywords may be used to define linkages.

FIx	<X>	<Y>	<R>	fix x-, y-displacement or rotation at node n
FRee	<X>	<Y>	<R>	opposite of FIx
Load	fx	fy	m	user-supplied directional forces or moments
Pin				establishes pin connection at node n
x,y				user-supplied x- and y-coordinates of node n

With these three commands, it is possible to specify structures of arbitrary geometry, boundary conditions, and properties.

6.2 Defining the Structural Element Geometry and Its Linkage to the Support Medium

The geometry and linkage must be defined for each beam or cable. A single beam or cable may be divided into a number of smaller segments to improve the accuracy of the calculation. The primary command shown earlier is used for this purpose. The keywords located within the brackets are optional choices, although one of each must be specified.

Beam elements are used for any structural member in which bending stiffness is important. Examples of applications for beam elements include:

(1) surface structures such as a foundation or footing (The logic used is general, so that any surface structure may be represented as an assemblage of beams);

(2) retaining structures;

(3) interior continuous tunnel supports such as a concrete or shotcrete liner or support such as steel or wood sets.

Examples of applications for cable elements include:

(1) tie-back anchors for retaining structures (point or fully-grouted anchorage);

(2) cable anchors (point or fully-grouted anchorage); and

(3) rockbolts (point or fully-grouted anchorage).

The beginning and ending linkage of the element is defined by the BEGIN and END keywords. Several options are available for each: (1) **grid=i,j**, (2) **node=n**, and (3) x_b,y_b or x_e,y_e.

The grid=i,j keyword denotes that the beginning (or ending) of the beam or cable is linked to gridpoint i,j of the rock or soil mass. The **node=n** command links the beginning (or ending) of the beam or cable to another node of the structure (i.e., another beam or cable element). By defining an x,y pair for the beginning (or ending) of the beam or cable, the user can place a structural element at any location within or outside the grid.

Each different type of beam or cable is assigned properties by using the **STRuct PROP=n** command. For example, if a beam is to be used (i.e., one structural cross-section), n=1, and the command may be

```
flac> struc prop=1  E=200e9  I=2.3e-5  c=4.8e-3
```

for a W6x25 beam (in SI units).*

*The Structural Steel Handbook lists sectional properties for standard beam sections.

It is noted that all quantities must be given in an equivalent set of units (see Table 4-1). The code does not take into account the weight of the structure when calculating loads. Therefore, if the weight of the structure is of importance in the problem, apply vertical forces to the structure equivalent of the weight.

The supplemental command, **struct node=n keyword** provides options for describing the type of connection between the structural elements. The options include:

(1) free or fixed x- and y-displacements or rotations;

(2) pin joints;

(3) applied loads or moments; and

(4) coordinate of the node

These linkage options are given by the following optional qualifying keywords.

FIx	X,Y,R	fixes the x- and y-displacements and/or the rotations at a given node point.
FRee	X,Y,R	frees the x,y displacement and/or rotation at a given node point.

Pin		provides a pin joint at a node (i.e., releases moments).
Load	fx,fy,m	applies an x- or y-directed load or moment at a node point. Loads are positive in the Cartesian directions. Counterclockwise moments are positive.
x,y		assigns coordinates to a new node number. Note that, if you already have created nodes using an option other than the **node=n** command, FLAC will automatically assign node numbers to node points as it creates them. To see the current node numbers, issue a **PRINT STRUCT** command.

6.3 Example Applications

A few simple examples are given to illustrate the use of the structural element commands in FLAC. The example in Section 2.0 illustrated a collapsing trench. Here, we support this trench with two struts which brace the excavation walls. The following command sequence is for the simple case where the braces are placed immediately upon excavation.

```
flac> grid 5,5
flac> m mohr
flac> prop s=.3e8  b=1e8  d=1600  fric=20  coh=0
flac> fix y  j=1
flac> fix x  i=1
flac> fix x  i=6
flac> set large
flac> hist nstep=1
flac> hist xdis  i=3  j=6
flac> set grav=9.81
flac> solve
flac> * excavate trench and install braces
flac> model null  i=3  j=3,5
flac> * properties for W6x25 beam in SI units
flac> struc prop=1 E=200e9 I=2.3e-5 area=4.8e-3
flac> struc beam beg gr=3,6 end gr=4,6 s=3 pr=1
flac> struct beam beg gr=3,4 end gr=4,4 s=3 pr=1
flac> step 50                  ┐ repeat to examine
flac> plot grid  beam          │ movement toward
flac> plot rf bound            │ equilibrium, plots
flac> plot bound dis beam      ┘ structure, reaction
                                 forces, displacements
```

Figure 6-1 illustrates the effect of the two braces (compared to Figs. 2-2 and 2-3). Trench collapse will still occur for this model but the failure region is reduced. Additional bracing and/or sheet piling (represented by vertical beams) may be tried to stabilize the trench).

A second example illustrates the loading of a foundation by a surface structure. Here, a simple cross-braced platform is constructed on a concrete slab which rests on a soil mass. The structure is loaded with vertical point loads on the supporting columns. The object is to examine the loads and moments in the structure as well as the stresses and displacements induced in the soil mass.

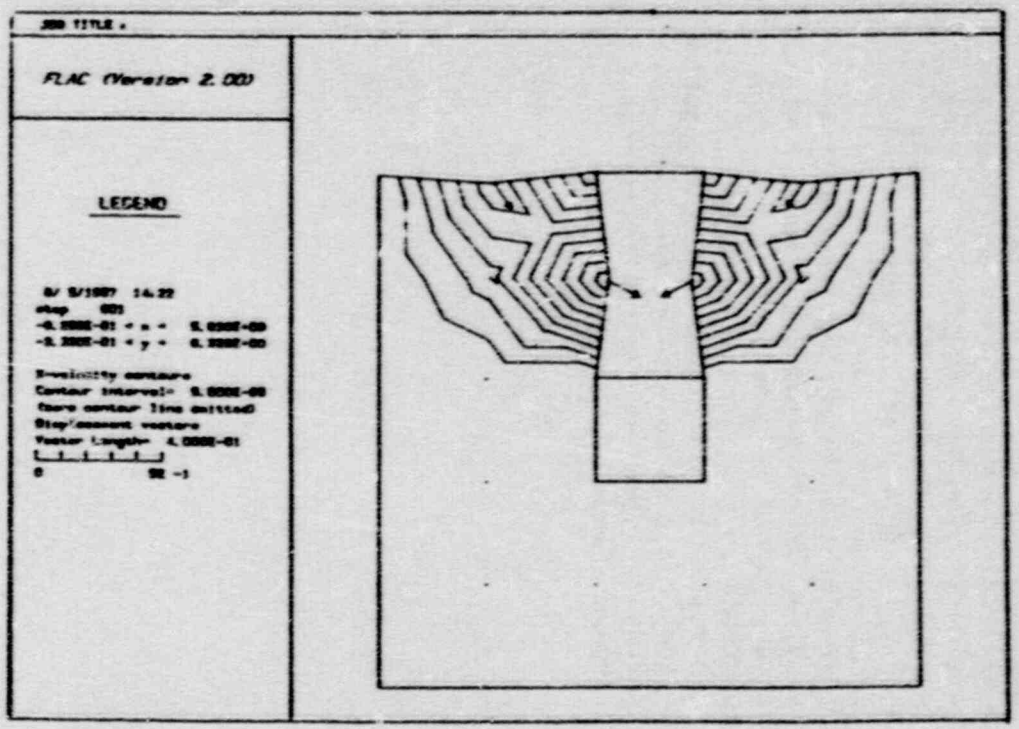


Fig. 6-1 Collapse of Braced Trench

The command structure below is used to set up and run this problem:

```
flac> * a simple cross-braced structure on a soil
flac> grid 10,10  * the soil mass
flac> m e
flac> prop s=.3e8  b=1e8  d=1600
flac> fix x  i=1
flac> fix x  i=11
flac> fix y  j=1
flac> set grav=9.81
flac> solve
flac> * let soil equilibrate under gravity
flac> * build structure
flac> * concrete slab
flac> struc prop=1  E=17.58e9  I=0.0104  a=.5
flac> struc prop=2  E=200e9  I=2.3e-5  a=4.8e-3
flac> struc beam beg gr 5,11 end gr 7,11 s=1 pr=1
flac> struc beam beg node 1 end 4,13 s=2 pr=2
flac> struc beam beg 4,13 end 6,13 s=2 pr=2
flac> struc beam beg 6,13 end 6,10 s=2 pr=2
flac> struc node=8 5.0,11.5
flac> struc beam beg node=8 end node=1 s=1 pr=2
flac> struc beam beg node=8 end node=4 s=1 pr=2
flac> struc beam beg node=8 end node=6 s=1 pr=2
flac> struc beam beg node=8 end node=2 s=1 pr=2
flac> struc node=1  fix r
flac> struc node=2  fix r
flac> struc node=4  Load 0 -1e6 0
flac> struc node=6  Load 0 -1e6 0
flac> plot beam * check structure
flac> pr struc * check linkage
flac> step 100                  ┐repeat to examine
flac> pr struc                  │movement to
flac> plot beam boun ydis       │equilibrium
flac> plot rf beam boun         ┘
```

The initial problem configuration is shown in Fig. 6-2. The effect of loading the soil with the structure is illustrated in Fig. 6-3. The forces and moments developed in the structure are a result of the applied loads and the interaction with the elastic grid are available by typing **PRINT STRUCT**.

A third example problem examines the loads developed in shotcrete and concrete liners for a circular shaft in a biaxial stress field. Here, beam elements are used to represent the lining in direct contact with the rock mass. Interface elements could also be used to simulate the effect of slip between the lining and the rock (see Appendix D).

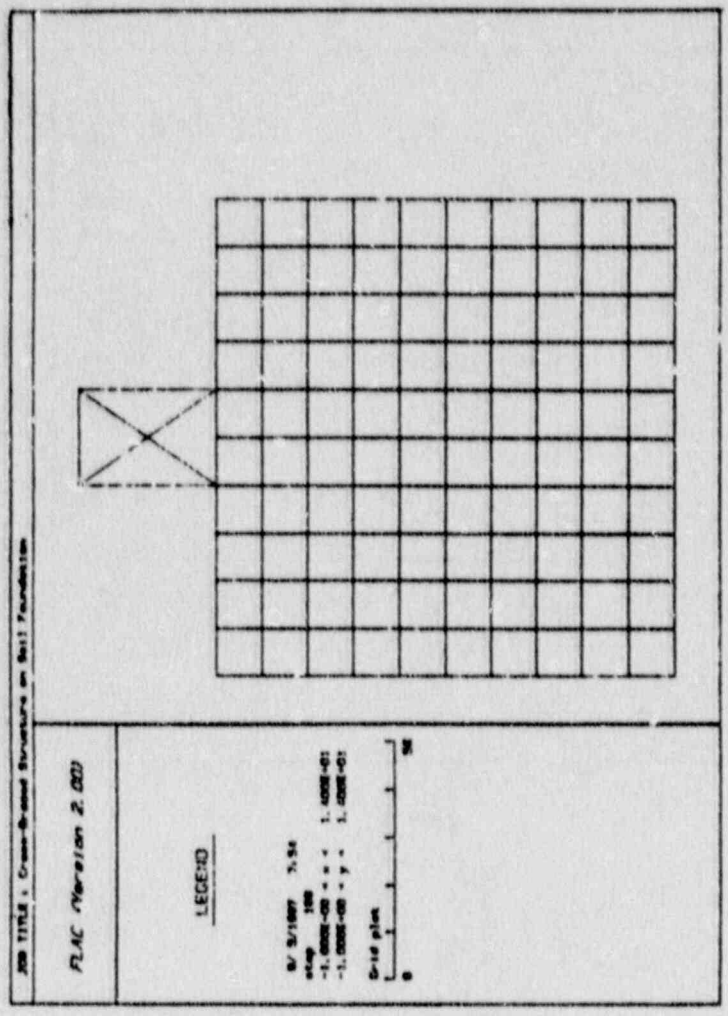


Fig. 6-2 Initial Grid and Braced-Beam Structure

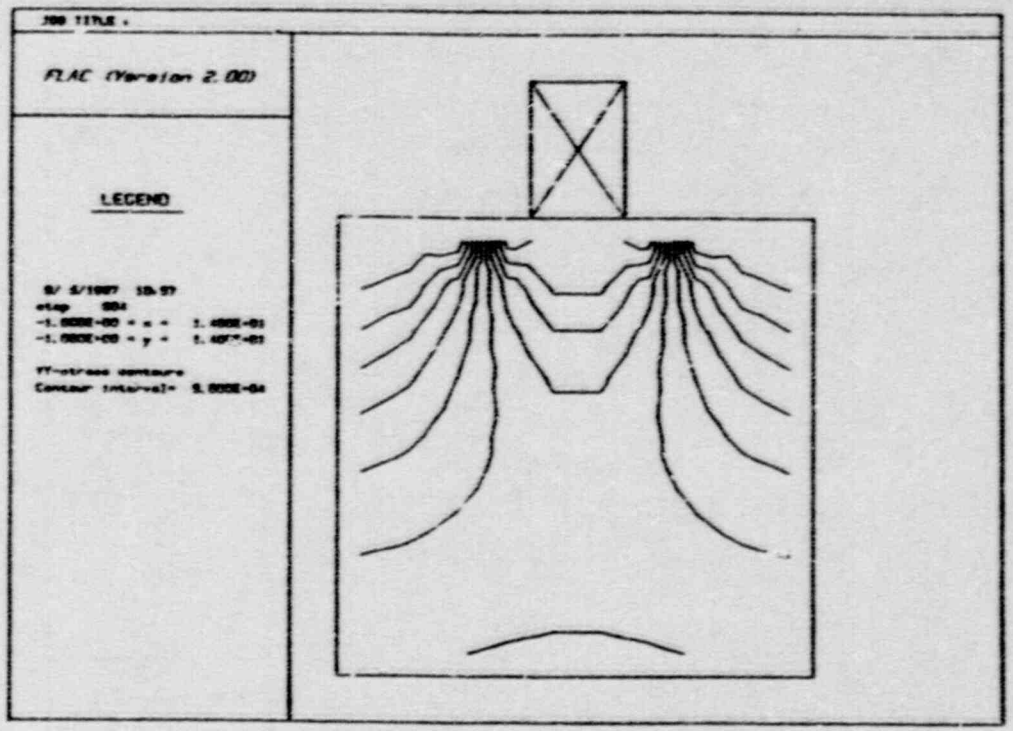


Fig. 6-3 Stress Changes Induced by Structural Loading

For this case, the development of the grid and liner is as follows:

```
flac> grid 15,15
flac> m mohr (Mohr-Coulomb model)
flac> gen 0,0 0,30 30,30 30,0
flac> gen 14,14 14,60 60,60 60,14 rat 1.2 1.2
          i=8,16  j=8,16 (stretch grid boundar-
          ies, leaving fine grid in center)
flac> gen 14,-30, 14,14 60,14 60,-30 rat 1.2 .833
          i=8,16  j=1,8
flac> gen -30,-30 -30,14 14,14 14,-30 rat .833
          .833  i=1,8  j=1,8
flac> gen -30,14 -30,60 14,60 14,14 rat .833 1.2
          i=1,8 j=8,16
flac> gen circle 14,14 4
flac> gen adjust
flac> * rock properties
flac> prop s=5.75e9 b=6.6e9 d=2000 coh=1e7 fr=35
flac> * boundary conditions
flac> fix x y  i=1
flac> fix x y  i=16
flac> fix x y  j=1
flac> fix x y  j=16
flac> ini sxx=-60e6 syy=-30e6 (initial stress)
flac> solve (at equilibrium; mine tunnel and in-
          stall support; then let it come to
          equilibrium)
flac> mod null region=8,8
flac> struc prop=1 E=13.8e9 I=2.8e-4 a=.15
          (concrete liner, 0.15m thick)
flac> pr mark
flac> struc beam beg gr 7 9 end gr 8 9
flac> struc beam beg gr 8 9 end gr 9 9
flac> struc beam beg gr 9 9 end gr 9 8
flac> struc beam beg gr 9 8 end gr 9 7
flac> struc beam beg gr 9 7 end gr 8 7
```

```
flac> struc beam beg gr 8 7 end gr 7 7
flac> struc beam beg gr 7 7 end gr 7 8
flac> struc beam beg gr 7 8 end gr 7 9
flac> plot grid beam
flac> step 100
flac> plo  bou  disp  red  beam
flac> pr struc
flac> pr state
```

repeat to examine movement to equilibrium

The loads and moments in the liner elements are listed (from **PRINT STRUCT**) in Table 6-1. The final stress state around the lined shaft is illustrated in Fig. 6-4.

Table 6-1

RESULTS OF PRINT STRUCT FOR TUNNEL LINER EXAMPLE

Structural node data ...

ID	x	y	x-disp	y-disp	ang-disp	xfix	yfix	rfix
8	1.000E+01	1.400E+01	1.955E-02	9.416E-05	-7.235E-05	no	no	no
7	1.117E+01	1.117E+01	1.279E-02	3.815E-03	-1.863E-03	no	no	no
6	1.400E+01	1.000E+01	3.403E-04	5.737E-03	-1.306E-04	no	no	no
5	1.654E+01	1.091E+01	-1.188E-02	3.793E-03	2.095E-03	no	no	no
4	1.800E+01	1.400E+01	-2.128E-02	2.984E-04	2.018E-05	no	no	no
3	1.683E+01	1.683E+01	-1.334E-02	-3.645E-03	-2.178E-03	no	no	no
2	1.400E+01	1.800E+01	5.952E-04	-6.271E-03	1.286E-04	no	no	no
1	1.091E+01	1.654E+01	1.434E-02	-3.678E-03	1.809E-03	no	no	no

ID	x-load	y-load	moment	i	j	pin	i	ang
8	0.000E-01	0.000E-01	0.000E-01	7	8	no		
7	0.000E-01	0.000E-01	0.000E-01	7	7	no		
6	0.000E-01	0.000E-01	0.000E-01	8	7	no		
5	0.000E-01	0.000E-01	0.000E-01	9	7	no		
4	0.000E-01	0.000E-01	0.000E-01	9	8	no		
3	0.000E-01	0.000E-01	0.000E-01	9	9	no		
2	0.000E-01	0.000E-01	0.000E-01	8	9	no		
1	0.000E-01	0.000E-01	0.000E-01	7	9	no		

Structural element data ...

ID	node-1	node-2	prop		F-shear	F-axial	moe-1	moe-2
8	8	1	1	beam	-3.087E+03	4.079E+06	-6.855E+03	-1.457E+03
7	7	8	1	beam	3.002E+03	4.073E+06	2.336E+03	6.855E+03
6	6	7	1	beam	-9.753E+01	8.277E+06	2.037E+03	-2.336E+03
5	5	6	1	beam	6.595E+02	9.351E+06	4.351E+03	-2.037E+03
4	4	5	1	beam	-3.917E+03	4.343E+06	-9.037E+03	-4.351E+03
3	3	4	1	beam	4.091E+03	4.520E+06	3.487E+03	9.037E+03
2	2	3	1	beam	-3.761E+02	9.382E+06	2.336E+03	-3.487E+03
1	1	2	1	beam	-2.568E+02	8.185E+06	1.457E+03	-2.336E+03

Structural properties ...

Prop no.	E	I	Area
1	1.380E+10	2.800E-04	1.500E-01

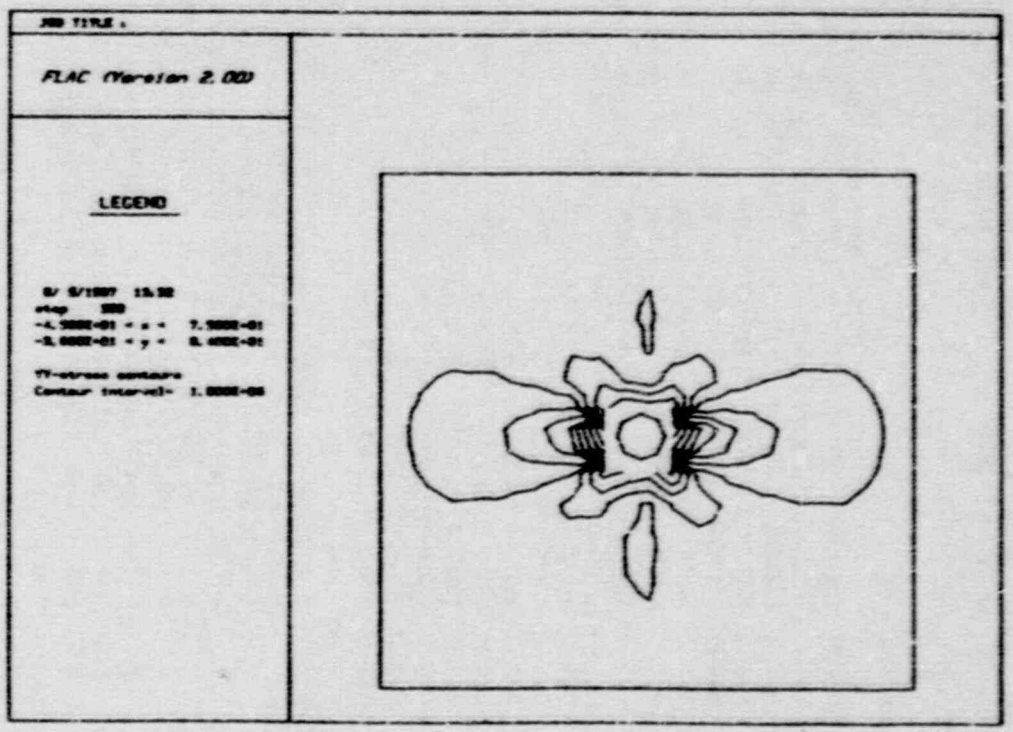


Fig. 6-4 yy-Stress Contours Around Lined Shaft

The last example problem examines the behavior of a lightly reinforced concrete beam subject to mid-point loading. Here, cable elements are used to represent the reinforcement. The input commands for this problem are:

```
flac> * a simple reinforced beam
flac> grid 10 2
flac> m e
flac> prop s=9e9 b=10e9 d=2400
flac> set large p-stress
flac> * boundary conditions
flac> fix y j=1 i=1
flac> fix y j=1 i=11
flac> apply yforce -1e7 j=3 i=6
flac> struc cable beg .1 .1 end 9.9 .1 s=12 prop=1
flac> stru prop=1 yi=1e6 kb=1e9 sb=1e7 e=200e9 a=2e-4
flac> his ydisp i=6 j=3
flac> solve f=1e5 s=1000 t=15
flac> save beam.sav
flac> ret
```

For this case, the vertical centerline displacement was about 12mm.

Figure 6-5 shows the reinforcing location superimposed on the x-displacement contours.

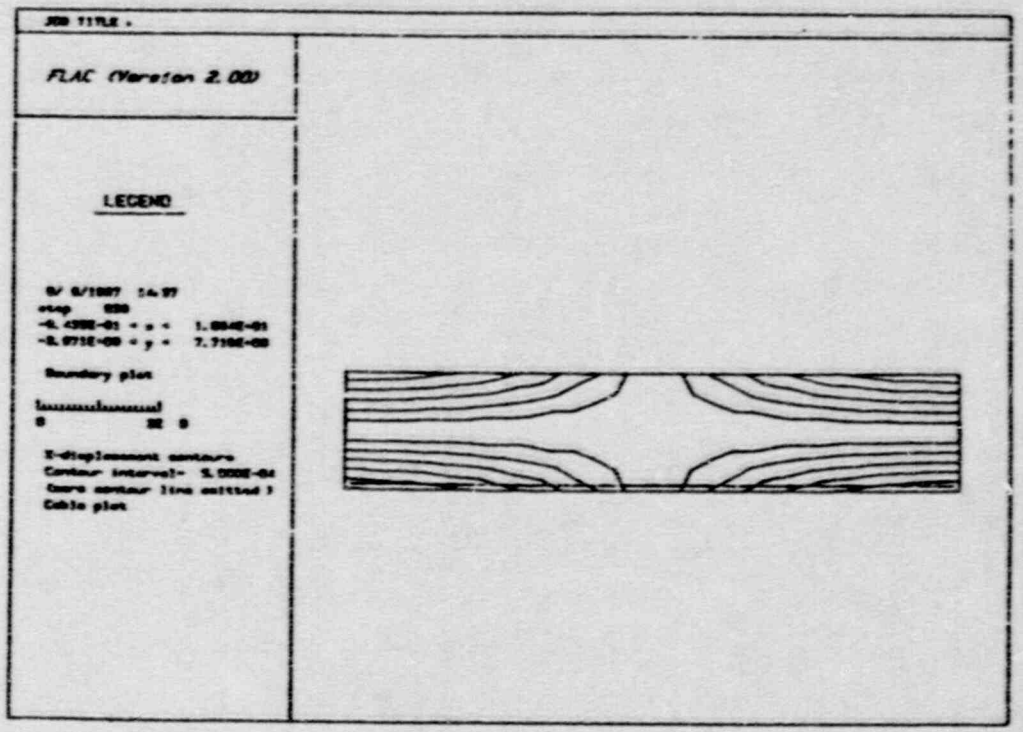


Fig. 6-5 Plot of x-Displacement Contours Superimposed on Reinforcing for Simply-Supported Beam Subject to Mid-Point Loading

7.0 EXAMPLE PROBLEMS

The following section provides a series of problems in the fields of soil and rock mechanics which have been chosen to illustrate the various features of the FLAC code. Where possible, the results are compared to analytic solutions for verification of code operation.

7.1 Example 1: Rough Footing on a Frictionless Cohesive Soil

A standard soil mechanics problem which can provide a rather severe test for the numerical model is the prediction of collapse load for a footing which bears on a cohesive soil (Sloan and Randolph, 1982). The soil-bearing capacity from the solution to "Prandtl's Wedge" is given by:

$$q = (2 + \pi)C$$

or

$$= 5.14C$$

where q is the load at failure and C is the cohesion of the material. The mode of failure is known to occur as shown in Fig. 7-1.

A rather coarse mesh was used for the FLAC simulation of this problem (Fig. 7-2). A 10-by-10 square grid is initially specified, but the right-hand and bottom boundaries are extended a bit to provide a reasonable approximation to a semi-infinite boundary. A symmetry boundary condition is imposed on the left-hand edge of the grid, whereas the bottom and right-hand edges are fixed in both the x- and y-directions. The footing load can be simulated by three possible methods:

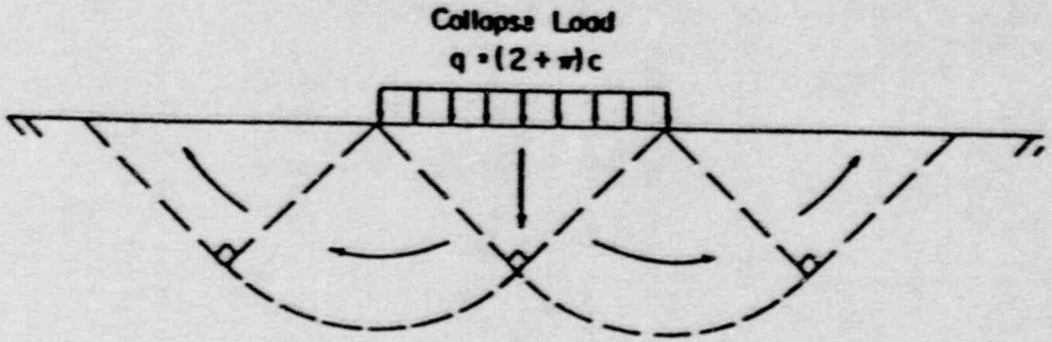


Fig. 7-1 Prandtl's Wedge Problem of a Strip Footing on a Frictionless Soil

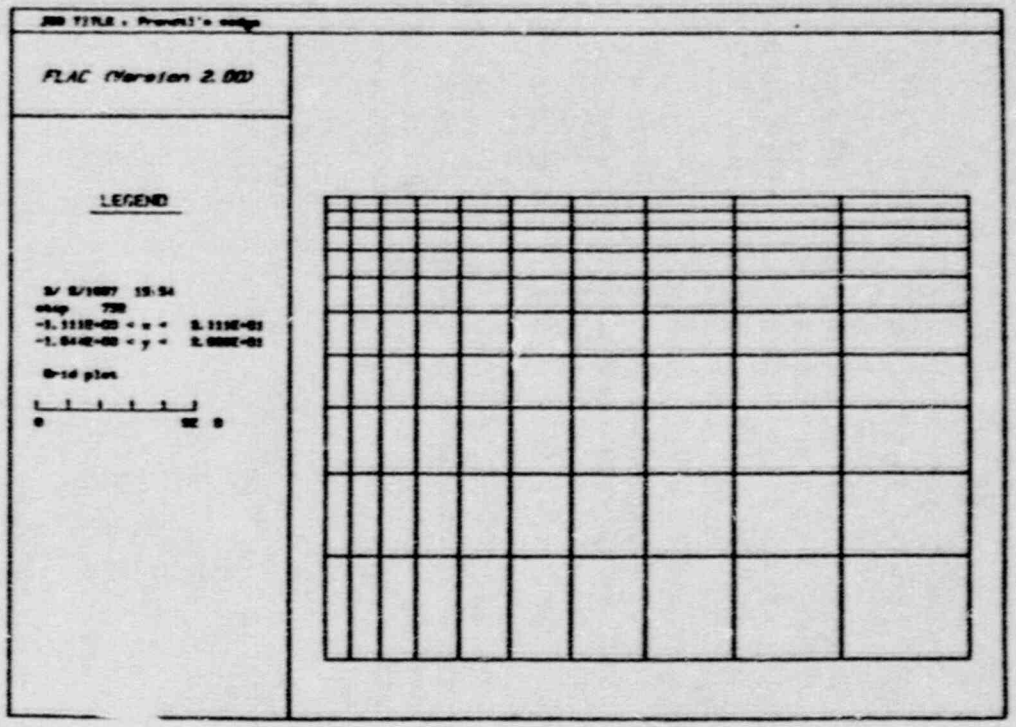


Fig. 7-2 Initial Grid Used in the Footing Example

(1) A pressure boundary condition may be applied in the negative direction across the desired surface elements.

(2) Gridpoint forces equivalent to the desired footing stress can be applied in the negative y-direction at surface grid points over the desired footing width.

(3) A constant velocity boundary condition may be applied in the negative y-direction across the desired footing width.

Because our aim, here, is to compare the predicted and exact solutions, the third alternative is chosen. The bearing capacity can be calculated by monitoring the force change in the grid beneath the boundary at failure. This eliminates the need for experimenting with applied force boundary conditions until collapse occurs.

The Mohr-Coulomb constitutive law was used for this simulation with the following material properties:

density	1,000.0 kg/m^3
shear modulus	0.3×10^8 Pa
bulk modulus	1.0×10^8 Pa
cohesion	1.0×10^5 Pa
friction angle	0

In addition, small strain logic is used for compatibility with the exact solution.

The input is given in the following command sequence. Note that the velocity condition representing the footing is obtained by using the **fix x,y (i=1,4, j=11)** command followed by initializing the y-velocity for these grid points to a value of -1×10^{-3} m/step.* Limits on the **SOLVE** command have been set to an out-of-balance force limit of 200 Newtons, a time limit of 40 minutes or a step limit of 1,000 timesteps. The time history of the y-displacement beneath the footing is kept and will be plotted at the conclusion of the run.

*When a **FIX** command is used, the normal assumption is that the velocities are fixed at zero. However, the velocities may be given any values, as illustrated above, using the **INITIAL** command. These will remain constant until changed.

```
set log on
tit
Prandtl's wedge
gr 10,10
*Mohr-Coulomb model
m m
prop s=.3e8 b=1e8 d=1000 fri=0 coh=1e5
*extend grid boundaries a bit
gen 0,0 0,15 20,15 20,0 rat 1.2 .8
*apply the boundary conditions
fix x i=1
fix x y j=1
fix x y i=11
*rigid footing moving at constant velocity
fix x y i=1,4 j=11
ini yv=-1e-3 i=1,4 j=11
*displacement history under footing
his nste =1
his ydis i=6 j=11
*solve with new limits
solve f=2e2 t=40 s=1000
set log off
save b:foot.sav
```

The run stops after 1,000 steps. By issuing the command

```
flac> print state
```

the following output will appear.

state

J	I 1	2	3	4	5	6	7	8	9	10
10	1.000	2.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	2.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
8	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
7	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.000
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.000
5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.000	0.000
4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.000	0.000
3	2.000	2.000	2.000	2.000	2.000	2.000	1.000	2.000	0.000	0.000
2	2.000	2.000	2.000	2.000	2.000	2.000	2.000	0.000	0.000	0.000
1	2.000	2.000	2.000	2.000	2.000	2.000	2.000	0.000	0.000	0.000

The values of 1 indicate that the material in that zone is currently at yield. The values of 0 and 2 indicate the material is currently elastic.* A better picture of the geometry of the yielded material can be obtained by plotting the displacement vectors and velocity contours. First, set the screen to the color mode. If you only have the monochrome screen, ignore the following two commands:

```
flac> set CGA (or set EGA)

flac> set pal=0
```

To create the plot, type

```
flac> plot xv  z  int=1e-4  yel dis  red  bo
```

The switch word **z** removes the zero contour interval and the specifier **int=1e-4** sets the contour interval to 1×10^{-4} m/sec.

Figure 7-3 will be reproduced on the screen. To obtain a pen plot (assuming you have properly connected the pen plotter—see the **PLOT** command), simply insert the word PEN after **PLOT** in the last command line.

---

*See the **PRINT STATE** command for more detail.

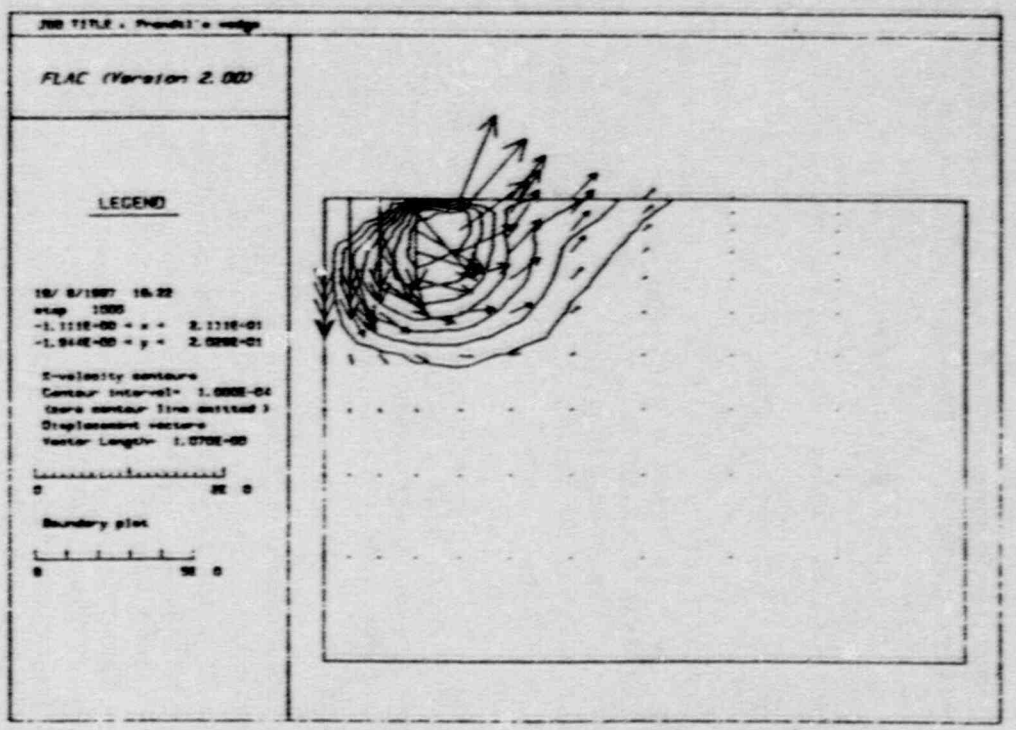


Fig. 7-3 Plot of Displacement Vectors and X-Velocity Contours for Footing Problem

The y-reaction forces beneath the footing at collapse are given by the command

flac> **print yreaction**

They can also be plotted by the command

flac> **plot rf**

The average stress at yield can be determined by summing the y-forces and dividing by the bearing area, yielding a value of q = 5.176×10^5 Pa, as compared to the exact solution of 5.14×10^5 Pa. Therefore, even the fairly gross discretization yields an error of less than 1% from the true solution.

The time history plot is shown in Fig. 7-4. The increasing displacement with time indicates the model has not come to equilibrium and is in active collapse.

A few final comments concerning the problem input are necessary. The solution limits to the **SOLVE** command have been re-defined from the default limits. Finally, note that this data file could easily have been set up on disk and called in by the program rather than with the interactive approach used here. The identical data set would be typed in, using an editor, to a disk file (e.g., FOOT.DAT) and called in by using the command

flac> **call foot.dat**

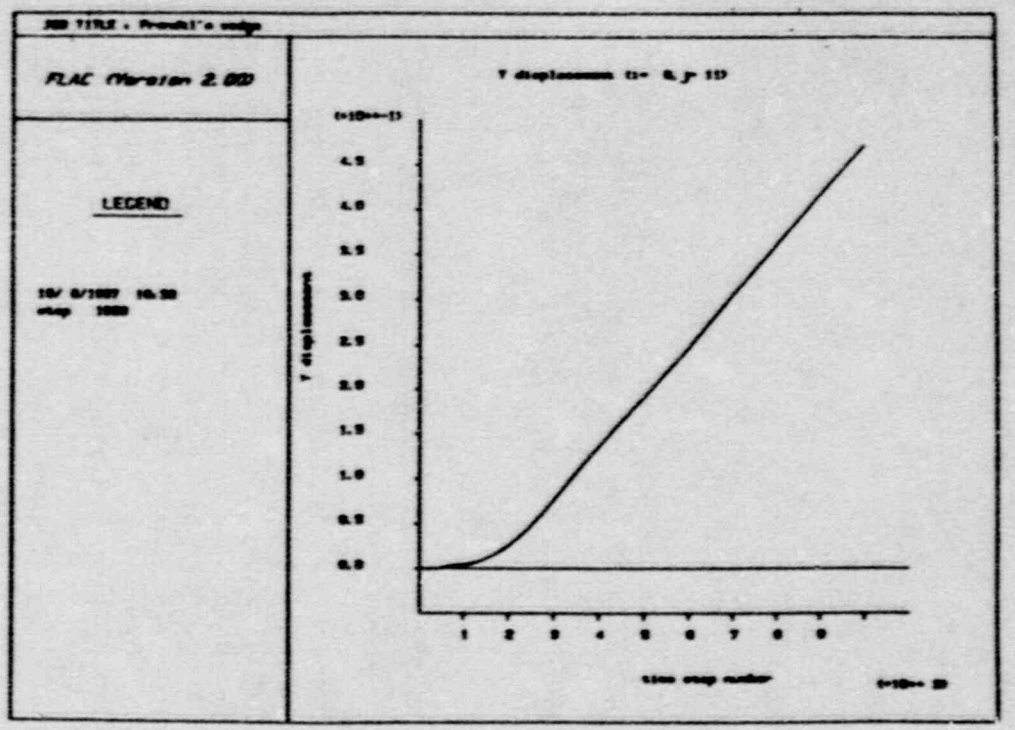


Fig. 7-4 Y-Displacement History Plot

The user will see each line executed in sequence as it is called in by FLAC. If you wish the information which is typed on the screen to be recorded in a file on hard disk or floppy, simply use the command

flac> **set log=on**

All data typed to the screen will be placed in a file FLAC.LOG on the hard disk.

Finally, the command **save foot.sav** will create a restart file of the current state of the simulation. To restart a saved state, simply use the **RESTORE** command on the default drive (e.g., flac> **rest foot.sav**).

7.2 Example 2: Slope Stability for a Cohesive and Frictional Soil

Another common problem encountered in engineering soil mechanics is the stability of soil slopes in frictional materials. In this example problem, we run two case studies. First, one in which a sand with zero cohesion is modeled with an initial slope steeper than the angle of repose of the sand. This slope, of course, should collapse, and we wish to plot the progression of this collapse as it develops. Second, we add a small cohesion to the material and re-examine it to determine if it is stable. We will use a simple example in which the soil is homogeneous. Here, the stability and factor of safety of the slope could be determined using an analytic or graphical technique. The power of the FLAC code lies in examination of more complex slope geometries in which, for exam-

ple, several layers of soil with differing material properties may exist. This type of problem can be examined with no greater effort by simply assigning differing material models to different zones.

This problem illustrates the use of several grid generation functions as well as zone "excavation" methods. The following command sequence is used to initialize the slope and create a restart file from which the boundary conditions and/or material properties may be varied.

```
;;;
*Granular slope under gravitational load
gr 20,10
*Mohr-Coulomb model
; a
*warp grid to form a slope - first use line command to
*form slope face
gen line 5,3 9,10
*mark area of slope in front of face so that two regions are formed
mark i=1,4 j=4
*null region in front of slope
and null reg=1,10
*soil properties - note large cohesion to force initial elastic
*behavior for determining initial stress state.  This will prevent
*slope failure when initializing the gravity stresses
prop s=.3e8 b=1e8 d=1300 fri=20 coh=1e10
*displacement history of slope
his ydis i=10 j=10
*displacement boundary conditions
fix x i=1
fix x i=21
fix x y j=1
*apply gravity
set grav=9.81
*solve for initial gravity stresses
solve
*reset displacement components to zero
ini xdis=0 ydis=0
*save initial state
save hill1.sav
*set cohesion to 0
prop coh=0
*use large strain logic
set large
step 200
save hill2.sav
step 200
save hill3.sav
return
```

An initial square grid of 20x10 zones is set up. Note that this initial grid will be assigned, by default, the dimensions of 20 units in x by 10 units in y unless otherwise re-defined by the user. A Mohr-Coulomb constitutive model is assigned to all zones (assumed because no range is given) with the following properties:

density	1,500 kg/m^3
shear modulus	0.3×10^8 Pa
bulk modulus	1×10^8 Pa
friction angle	20°
cohesion	1×10^{10} Pa

You will notice that we have assigned a very large value to the cohesion for this slope initially. The reason for this can be seen by re-examining the way in which an explicit model works. We must form an initial grid and, in this case, apply gravity to the grid points and allow the gravitational stresses to equilibrate. Because the model is dynamic in nature, we step through time, consolidating the material until the velocities of the grid points are small. For most problems, we wish to have this consolidation process occur as rapidly as possible. This can be done by requiring the material to behave elastically during the consolidation. Once stresses have equilibrated, we may assign whatever material properties we choose, excavate, apply loads, etc., and continue the simulation process. In the case illustrated here, we assign a plastic constitutive model initially but set the cohesion high, forcing the ma-

terial to behave elastically. Once the stresses have equilibrated, we can simply re-define the cohesion to be the desired value. This eliminates having to re-define the constitutive model and properties after equilibrium, thus saving a few steps of input. The same effect could be generated by using an initial elastic model for consolidation of the grid, followed by a change to the Mohr-Coulomb model, as well as definition of the material friction and cohesion. Either methodology for initializing a gravity-loaded grid is acceptable and is determined only by the preference of the user. It is important, however, that the user follow a general solution procedure as illustrated in Fig. 5-1.

We have defined the basic grid, but it must now be deformed into the shape of a slope and the soil base beneath it. This is done using the **GEN** command.

First, we define an internal line which we will use to define the eventual slope boundary. This is done with the command

flac> **gen line 5,3 9,10**

A line is generated beginning at (x,y) = (5,3) and ending at (x,y) = (9,10). You may wish to plot the grid to verify that an internal line has been generated. Note that the given endpoints of the line should coincide with existing gridpoints if the full line is to be generated. You can reposition gridpoints (with the **INI** command) prior to using **GEN** if they are not already in the correct positons. We next null out the area directly to the left of slope face without eliminating the

slope toe. This is done by creating a "region" (i.e., we need to divide the grid into two regions separated by a boundary). You will note that the line which defines the slope face starts within the grid and must be extended to the left-hand boundary. Two methods are possible for extending this boundary. Another **GEN LINE** command could be used, extending along a horizontal line from the left edge of the grid to the end of the line, or the same can be "marked" to notify FLAC that a boundary exists. Here, we will use the **MARK** command. The entry

```
flac> mark i=1,6  j=4
```

marks a boundary line along j=4 from i=1,6. Now, we have successfully defined two regions: the slope and the area to be removed ahead of the slope face. This can be verified by issuing the **pr mark** command. This area may be removed by typing

```
flac> mod null  region=1,10
```

which will null out the region in which zone 1,10 lies. Now, plot the grid to see the final result.

The acceleration due to gravity is next set to 9.81 m/sec^2 (positive means acting downward). The boundary conditions applied include roller boundaries on the front and rear of the model as well as a fixed base. Now, we want to obtain an initial elastic state in which gravitational stresses are equilibrated. This is done by using the **SOLVE** command with the default limits. Equilibrium is obtained when the out-of-balance force limit of 100 Newtons is reached. As a means of examining the progression of the run, the user may

request the y-displacement history at a gridpoint at the peak of the slope. This is done using the **HIS** command. When the **SOLVE** command has reached its limit, try plotting the history to verify that the mesh is, indeed, at an equilibrium state (i.e., **plot his 1**). A numerical and graphical representation of the elastic stresses can be obtained by issuing the following commands:

```
flac> print sxx syy

flac> plot sxx  yellow  bound*

flac> plot syy  yellow  bound*
```

At this point, it would be smart to create a restart file which will save the present equilibrium state. This is done to save time in case future runs will be made in which material parameters or constitutive models are to be varied. Performing these studies requires only that the elastic state be restarted, therefore eliminating the necessity to re-develop the equilibrium state.+ The command

```
flac> save hilll.sav
```

---

*assumes screen initialized to color screen, palette 0 prior to plotting

+It is worthwhile mentioning here that, particularly when complex excavation sequences or material properties variations are to be modeled, liberal use of the restart option is suggested.

will create a restart file on the default drive called HILL1.SAV. We could quit FLAC at this point and restore the saved state at a later time simply by typing

```
flac> restore hill1.sav
```

At the current timestep, the simulation is still in RAM, so we can proceed from here.

For the next stage of the simulation, we want to set the material properties to the actual soil values and time-step while examining the possible failure process. During this process, plots of the progressive displacement of the slope will be made. To avoid any confusion in analyzing the data, we prefer to see only the change in displacement and not the cumulative displacements since the beginning of the simulation (i.e., time-step = 1). The explicit procedure does not require displacements in the calculation process but keeps the cumulative total for each grid point as a convenience to the user. Therefore, we may zero them out (i.e., initialize) at any point in the calculation procedure without affecting the resulting answers. This is done by using the command

```
flac> ini xdis=0  ydis=0
```

From this point on, plots or print-outs will show only the change in displacement from the previous state.

Next, the material properties of the zones are reset by using the **PROP** command. The cohesion is set to zero for all zones which are currently composed of soil. This is done by using the **REGION**

command once again. Finally, we set the calculation mode to LARGE STRAIN to provide a more accurate geometrical representation of the slope failure as it progresses. Because slope collapse will occur due to the angle of repose of the soil, we do not want to use the **SOLVE** command (equilibrium will not be reached).* It is easier to use the **STEP** command and simply time-step the simulation a small number of steps at a time, stopping to print and plot these intermediate stages. Here, we see the power of the explicit method in its ability to follow highly non-linear problems, which may never converge to an equilibrium state, through time.

Try stepping in 200 timestep intervals, followed by printing and plotting the results, by issuing the command

```
flac> step 200
```

followed by

```
flac> print xv  yv  xd  yd  state

flac> plot xv  z  yellow  displ  red bound
```

The following series of figures show the results of the plot commands given above at 75, 475 and 1275 timesteps. These figures clearly illustrate the progressive collapse of the slope.

---

*Of course, the **SOLVE** command could be used if larger out-of-balance force limits or a smaller number of timesteps or total run time were imposed.

Figure 7-5 shows the initial grid plot which was produced by the command

flac> **plot pen grid** (plots grid with auto window)

The problem is then stepped 200 additional timesteps by typing

flac> **step 200**

Figure 7-6 shows the x-velocity contours superimposed on the displacement vectors and slope boundary at timestep 475. This figure was produced by the command

flac> **plot pen xv z red disp max=0.4 gre boun gre**

You can clearly see the beginnings of the slope collapse. The x-velocity contours illustrate the development of the yielding face. Figure 7-7 was plotted after another 800 timesteps. Here, the slope is collapsing in an attempt to reach its angle of repose. Obviously, the displacements of the grid points become unrealistic (this is a continuum code), with continuing timestepping. FLAC automatically checks for excessive grid deformation and will stop the calculation process if it is detected, displaying an error message without loss of the run.

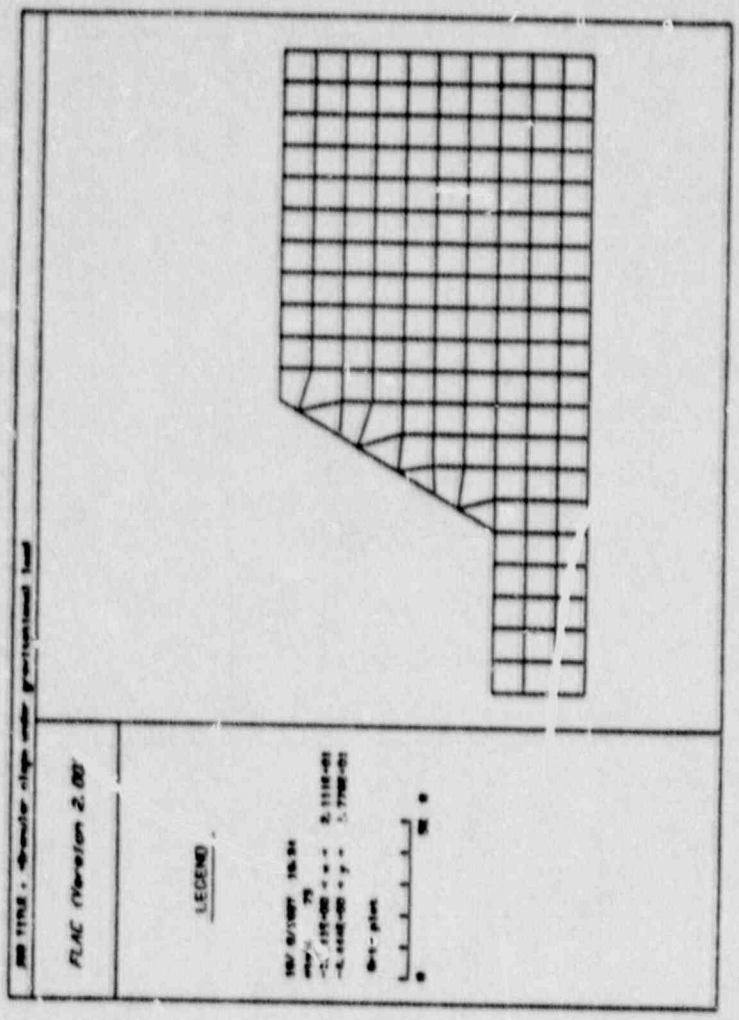


Fig. 7-5 Grid Plot of Slope at Initialization

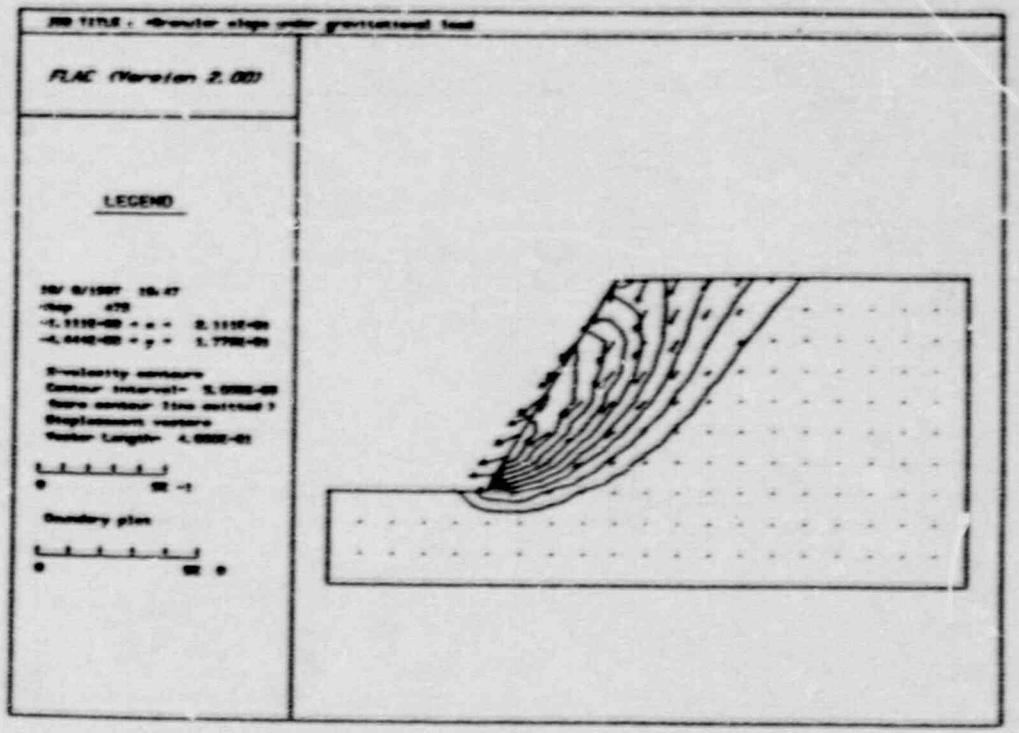


Fig. 7-6 Plot of Displacement Vectors and X-Velocity Contours at Step 475

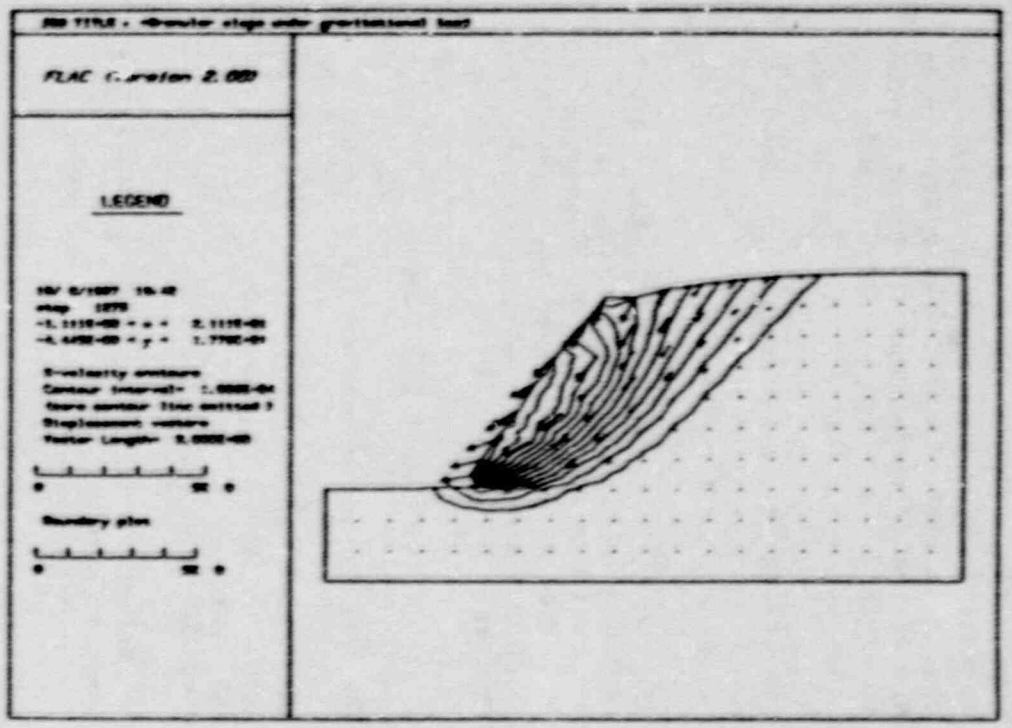


Fig. 7-7 Plot of Displacement Vectors and X-Velocity Contours at Step 1275

The user must realize that there is a limit to which the large distortions are meaningful. As the shape of the zones becomes distorted to too great an extent, the resulting calculations may become inaccurate.

After completing the above simulation, type

```
flac> new
```

and a new run will be initialized. An interesting comparison can be obtained by running another simulation in which a small cohesion is assigned to the soil. Because we have saved the initial elastic equilibrium state, we need only restart from this timestep:

```
flac> rest hill1.sav
```

Following the procedure used earlier, type

```
flac> ini xdis=0  ydis=0
flac> prop coh=7x10^4  region=10,1
flac> set large
flac> step 200
```

Run this problem as before, examining the effects of a change in properties.

7.3 Example 3: Elastic Cantilever Beam with End Shear

The following example analyzes the bending of a cantilever beam under the action of a shearing force across its end surface (Fig. 7-8). Although continuum codes such as FLAC are not best suited to analyzing structural problems involving bending moments, this problem presents a fairly difficult validation test.

The input data given is given in the following command sequence.

```
set log on
tit
Beam with end shear
gr 30,6
*
*5:1 beam with 6 zones across width
*
* *
set p-stress col=132
prop d=1000 bulk=1e8 shear=.5e8
*clamp end of beam
fix x y i=1
*apply end shear
apply yforce=-0.142857e5 i=31 j=1,7 *(load of 1e5 units)
step 3500
set log off
save colan.sav
return
```

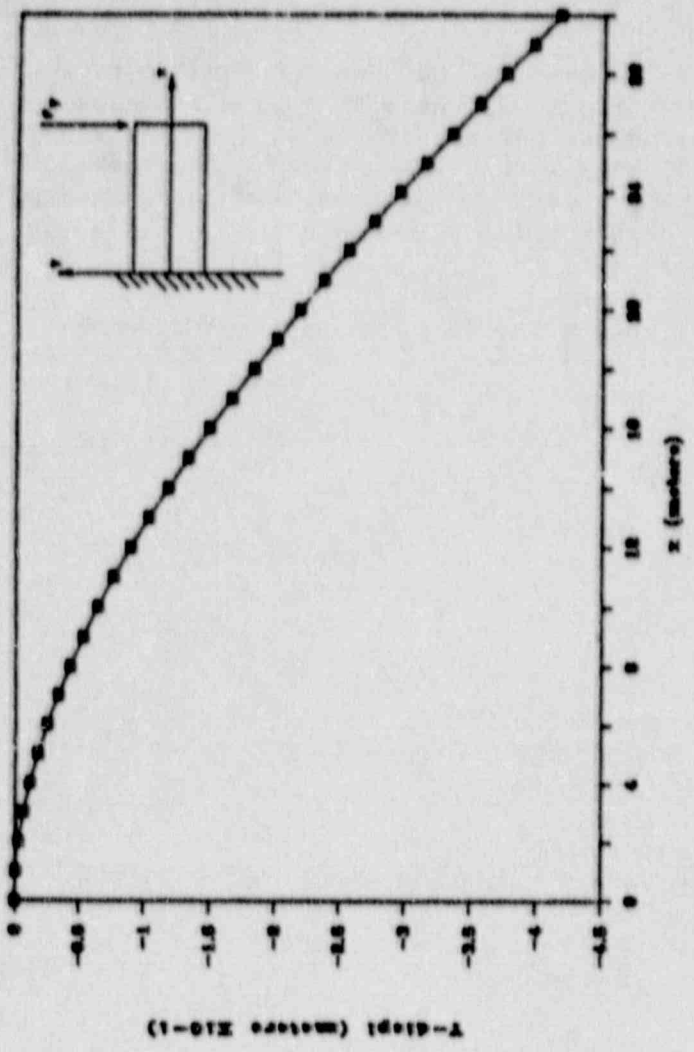


Fig. 7-8 Cantilever Beam Configuration and Y-Displacement of Beam Centerline

A grid of 30x6 zones represents the beam in plane stress. The output is also set to 132 columns. One end of the beam is fixed in x and y, while the rest remains unconstrained. A negative y-force of 20×10^5 Newtons is applied to the opposite end of the beam. This load is distributed among the seven end grid points for a load per grid point of -0.142857×10^5 Newtons. Although this example runs rather slowly, convergence is obtained after about 3500 timesteps. Figure 7-8 illustrates the final y-displacement along the beam.

The final displacement is 0.41 m, compared with an analytic solution of 0.39 m--or an error of approximately 5%.

7.4 Example 4: Circular Tunnel In an Elastic, Elasto-Plastic, and Transversely-Isotropic Rock Mass Under Initial Stresses

This example illustrates the use of several constitutive models as well as the **GEN** commands to generate a mesh coordinate system and the circular boundary of a tunnel in an infinite rock mass in plane strain. The method of introducing symmetry conditions into the model is shown by examining only one-quarter of the hole. The use of the initial stress command and the application of boundary pressures to provide in-situ field stresses are also illustrated in this example.

7.4.1 Elastic Rock Mass

— The input data for this problem is given in the following listing. The file can be set up on disk and read in by using the **CALL** command, or it can be typed in interactively.

```
tit
Hole in an elastic aedius
* generate the grid
gr 20,20
*elastic aodel
a e
*give initial grid coordinates and grade aesh to boundaries
gen 0,0 0,6 6,6 6,0 rat 1.2 1.2
*generate a quarter circle in lower left hand corner of grid
*using the arc function
gen arc 0,0 1,0 90
*elastic properties
prop s=2.9e9 b=3.9e9 dens=2500
*initial stresses
ini sxx=-30e6 syy=-30e6
*boundary conditions for a quarter syaaetry problea
fix x   i=1
fix y   j=1
fix x y i=21
fix x y j=21
*aonitor history of displaceaents at tunnel periphery
his nste=1
his ydis i=1 j=12
his xdis i=12 j=1
*aolve for initial stress state
solve
*now excavate hole
aod null region=1,1
*tiae step to equilibriua
step 500
*create a save file
save ahole.sav
*return to interactive aode
return
```

A 20-by-20 zone grid (400 elements) is specified with elastic constitutive model. We wish to have a tunnel radius of 1 meter and have the tunnel periphery be finely discretized so that the results are reasonable. To obtain this, we would like to have an x- and y-grid spacing of about $1/5^{th}$ the radius in the area encompassing the tunnel, but we also need to have the boundaries about 5 radii away in order to avoid influencing the displacements at the hole. To create an initial grid spacing of 0.2m, the **GEN** command is used to re-define the mesh coordinates to 0-6 meters in the x- and y-directions. If you wish to confirm the results of **GEN**, use the **print x** or **print y** command.

The tunnel periphery is formed by using the **GEN ARC** command. The effects of this command can be viewed by setting a window to

```
flac> wind -.5,2  -.5,2
```

and by plotting the grid. After the initial stresses have equilibrated in the model, we will excavate the zones interior to the tunnel periphery.

Next, the boundary conditions must be set to represent symmetry conditions along the left-hand and lower boundaries and fixed displacement boundaries set at "infinity" (the top and right-hand boundaries). This is done by placing rollers along the i=1, j=1,21 (left) and j=1, i=1,21 (bottom) boundaries and fixed points along the i=21, j=1,21 (right) and j=21, i=1,21 (top) boundaries. With the grid now fixed in space, in-situ stresses can be applied using the **initial sxx=<value> syy=<value>** command.

As was the case in the previous example, these stresses must first be equilibrated prior to excavation (as is the physical reality). This requires only one timestep, since FLAC simply loads the stresses into proper positions in its main array. This can be accomplished with either of the following commands:

```
flac> step 1
```

or

```
flac> solve
```

You can verify that the proper initial stress state was obtained by using the **print sxx** or **print syy** command*. At this point, it is best to make a save file which can be restarted from this initial state.

Next, the excavation is created by "nulling" out the tunnel zones. This is done with the series of **model null** commands. To verify that you have nulled the proper zones, plot the grid.+ The entire grid and a blow-up of the excavation are

---

*There are several alternative methods of inducing initial stress in the grid. You can apply boundary pressures or boundary loads which will produce equivalent internal stresses.

+If a mistake was made, use the **NEW** command to start over and restart the initial state.

shown in Figs. 7-9 and 7-10. These plots were produced with the commands

```
flac> plot pen grid * plots full grid
flac> wind -.5,2 -.5,2 (creates small window)
flac> plot pen grid (plots grid around the excavation)
```

Now, the problem is stepped to equilibrium. For a 20x20 grid with small strain logic, approximately 30 sec/timestep is required on the standard PC with the slow 8087 (4.7 MHz) chip. Thus, 100 timesteps will require roughly 1 hour of solution time. The elastic solution here will require approximately 500-800 steps to arrive at the equilibrium solution (depending on your definition of equilibrium).

It is good practice to make history plots of the displacement at points undergroing the maximum change (i.e., at an excavation boundary) to determine the progression of the model toward equilibrium. You will quickly be able to see the convergence characteristics of the problem. You may wish to plot the stresses and displacements on the screen using the **PLOT** command. Recall that if you wish to use the color screen, you must type **set xga** (or **set ega**) and use whichever palette choice you prefer.

A comparison of the results to the analytic solution (Kirsch solution) for a hole in an infinite plate in plane strain is shown in Fig. 7-11. As you can see, the solutions compare reasonably well. The error is attributed to the rather course zoning adjacent to the tunnel. A similar run with a 30x30 zone model produced an agreement within 5%.

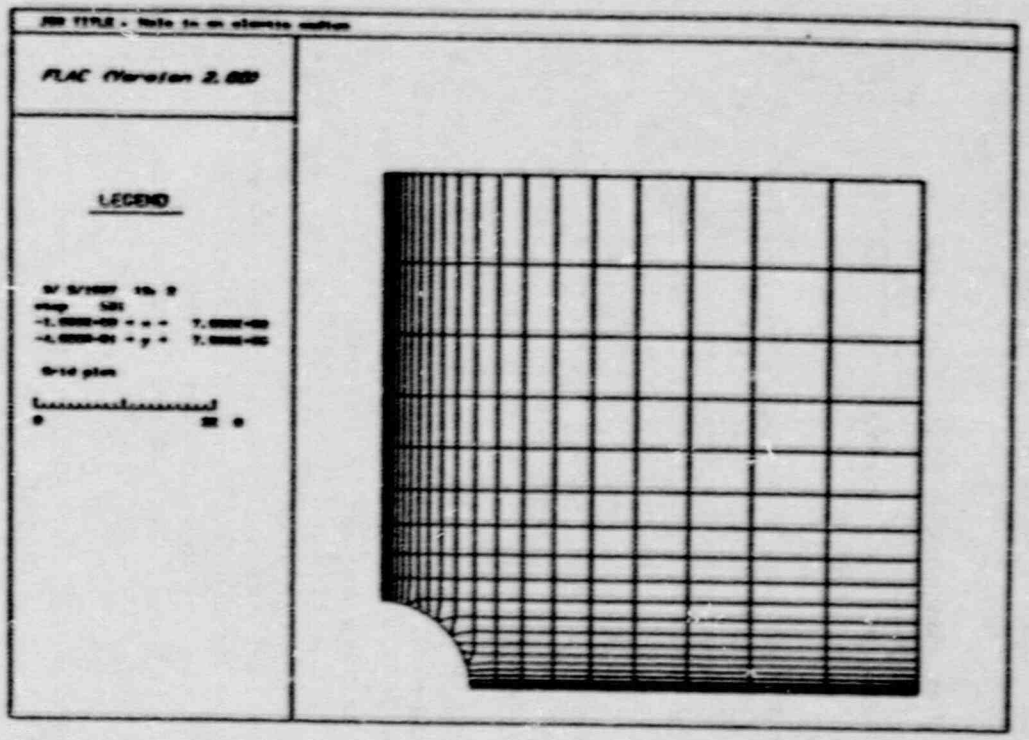


Fig. 7-9 Plot of the Grid Used in Tunnel Simulation

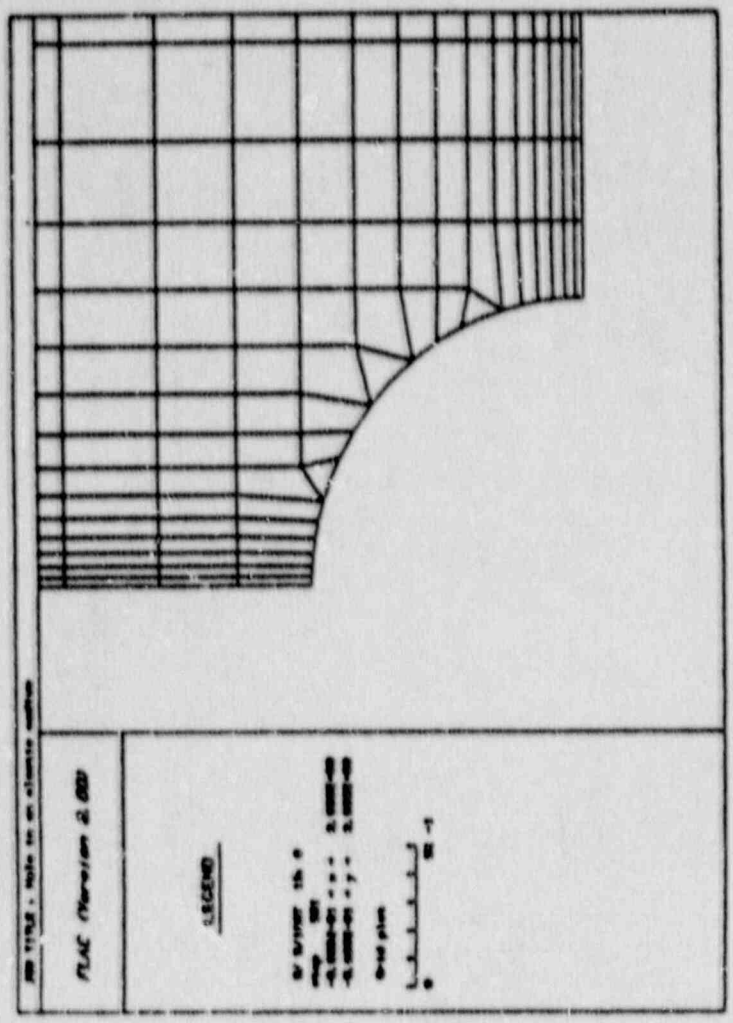


Fig. 7-10 Close-Up View of Tunnel Discretization

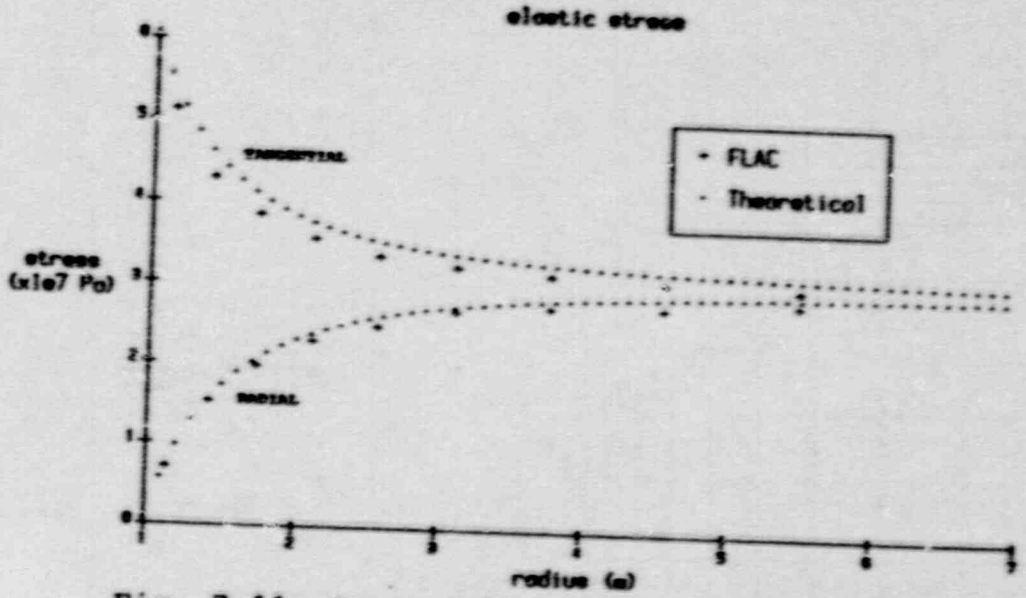


Fig. 7-11 Comparison of Tangential and Radial Stress from FLAC to the Kirsch Solution

7.4.2 Elasto-Plastic Rock Mass — Next, a comparison is made to the same problem but using an elasto-plastic constitutive law. The only adjustments necessary to the input data shown earlier are new constitutive law and property definitions. This is done by changing the second and third lines of the input data file to read:

```
flac> model mohr

flac> prop s=2.8e9 b=3.9e9 d=2500 fric=30
      coh=3.45e6
```

The **SOLVE** command is used to bring stresses to equilibrium and requires one timestep. The tunnel is mined using the previous **model null region** command. Finally, the displacements can be reset using the **initial xd=0 yd=0** command. For the simulation to come to equilibrium, approximately 500 to 600 timesteps are necessary—or about 2-3 hours of solution time on the standard IBM PC. Times will be significantly less on the AT or equivalent with higher speed co-processor (6-12 MHz) or when using one of the commercially-available enhanced speed processor boards.

The stress profiles around the opening for the plastic case are shown in Fig. 7-12. This figure clearly illustrates the yield zone which has formed around the tunnel, resulting in the peak tangential stress moved to a radial distance beyond its limit. The theoretical and predicted profiles match the theoretical solution (Bray, 1967) quite well. The radius of the broken zone is given by Eq. (7-1).

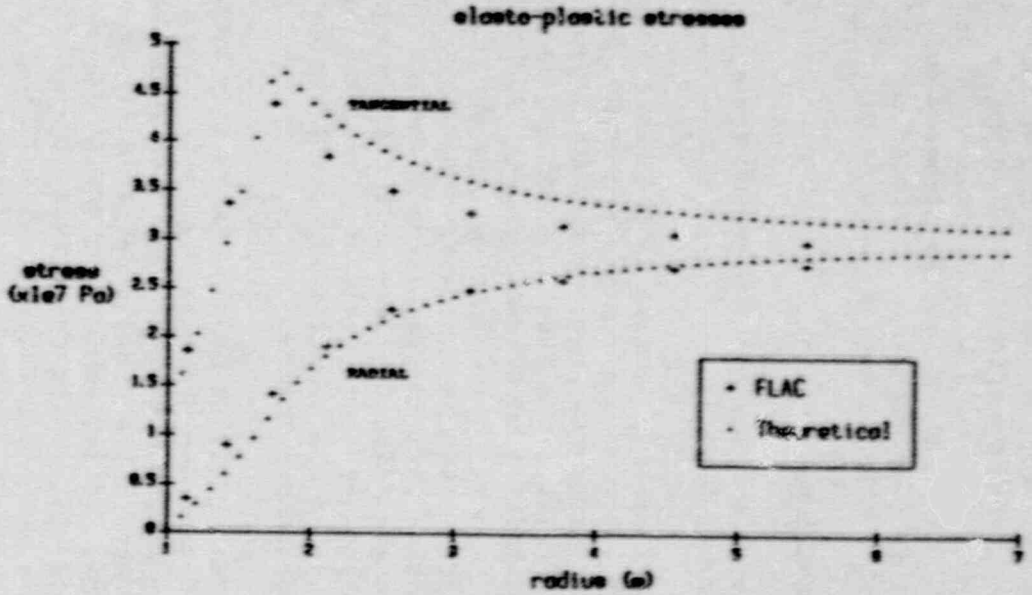


Fig. 7-12 Comparison of Tangential and Radial Stresses from FLAC and Bray (1967)

$$r_{br}=a\left[\frac{2p-q_u+[1+\tan^2(45+\phi/2)]\ C\ \cot\phi}{[1+\tan^2(45+\phi/2)]\ C\ \cot\phi}\right]^{1/Q} \qquad (7\text{-}1)$$

where a = tunnel radius

C = cohesion

p = far-field hydrostatic pressure

$$Q=\frac{\tan(45+\phi/2)}{\tan(45-\phi/2)}-1$$

q_u = uniaxial compressive strength

ϕ = friction angle

The tangential and radial stresses in the plastic zone are given by

$$\begin{aligned}\sigma_\theta &= C\ \cot\phi[(r/a)^Q-1]\\ \sigma_r &= C\ \cot[Q(r/a)^Q-1]\end{aligned} \qquad (7\text{-}2)$$

and the stresses in the elastic zone are given by

$$\begin{aligned}\sigma_\phi &= P+(\beta/r^2)\\ \sigma_r &= P-(\beta/r^2)\end{aligned} \qquad (7\text{-}3)$$

where $\beta = \left[\frac{[\tan^2(45+\phi/2)-1]p + q_u}{\tan^2(45+\phi/2)+1} \right] r_{br}^2$

In this case, the radius of the broken zone using Eq. (7-1) is 1.85 meters. You can examine the extent of the yield zone predicted by FLAC by typing

```
flac> print state
```

or

```
flac> plot state
```

The state indicators show that the yield zone is approximately 1.6 m and is in reasonable agreement with the analytic solution. Again, the error is attributed to the relatively course grid. A 30x30 zone model agreed within less than 7%.

The principal stress vectors, displacement vectors and shear stress contours for this problem are given in Figs. 7-13, 7-14 and 7-15. These plots were produced by the following command set.

```
flac> set cga pal=0 (if you have a color monitor)
flac> wind -.5,2 -.5,2
flac> plot pen str bou
flac> plot pen disp max=.3 red bou green
flac> sclin 1  0,0  5,5
flac> plot pen szy z boun green
```

If you wish to have only a screen plot, eliminate the keyword PEN.

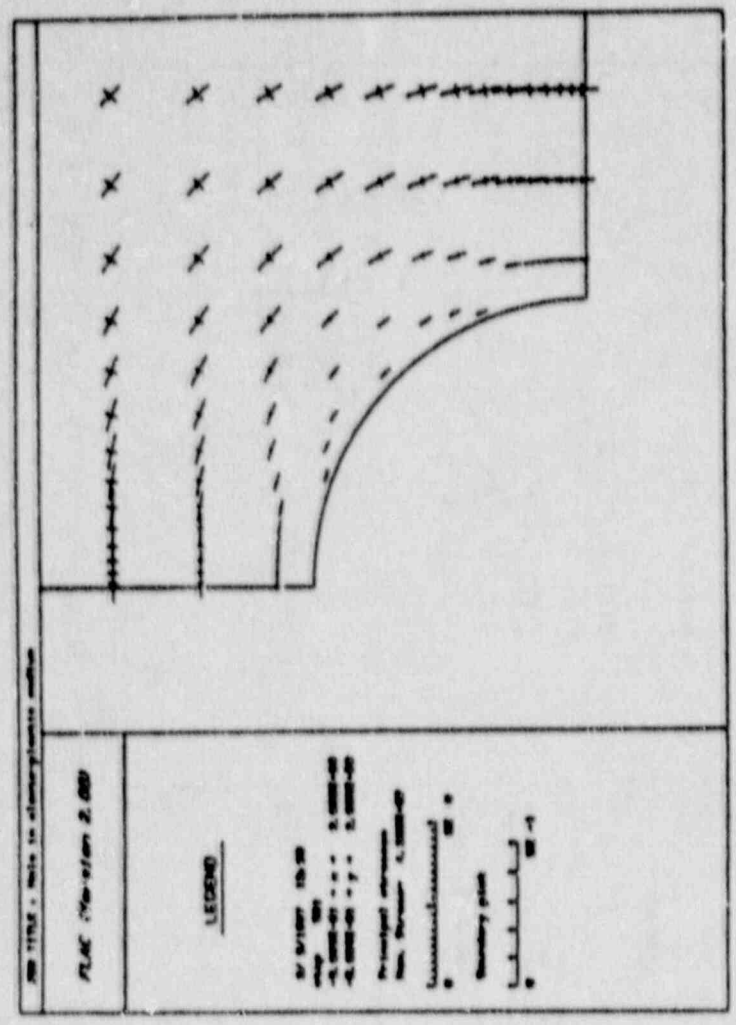


Fig. 7-13 Principal Stress Vectors for Elasto-Plastic Example

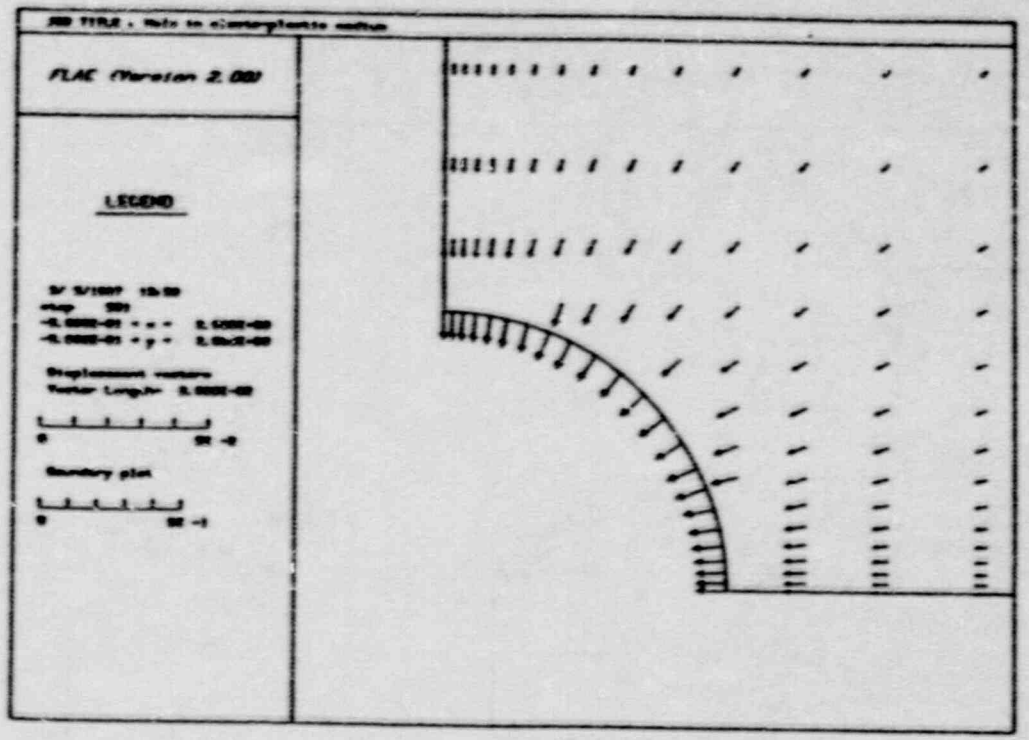


Fig. 7-14 Gridpoint Displacements for Elasto-Plastic Example

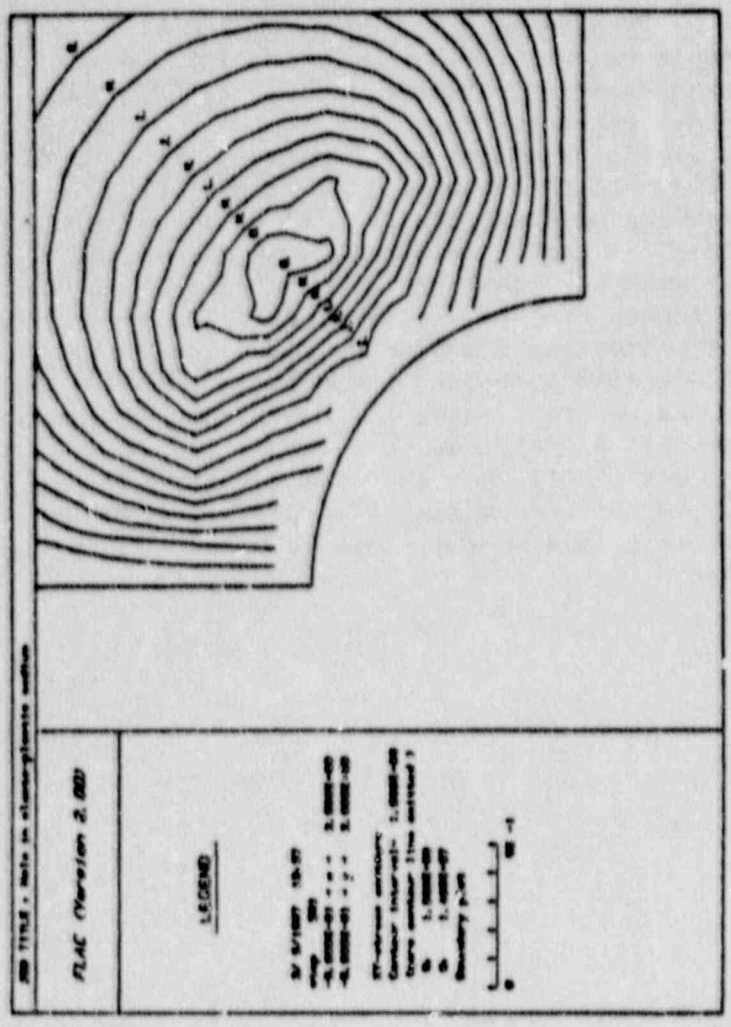


Fig. 7-15 Shear Stress Contours for Elasto-Plastic Example

7.4.3 Transversely-Isotropic Rock Mass — A final example calculation is performed for the tunnel in a transversely-isotropic rock mass under initial biaxial stresses (Fig. 7-16). The solution for the displacements of the tunnel is given by Eissa (1980). In this example, the use of boundary pressures to bring the initial stresses to equilibrium is shown. We start with an elastic rock mass with the identical grid as used in the previous tunnel problems. The input data is identical to the previous examples with these exceptions: the material model and properties must be changed and, rather than using a fixed boundary, boundary pressures are applied to maintain in-situ stresses. Note that the symmetry here is valid only in that the angle of anisotropy is 0: if it were at some some angle, the symmetry is not sensible.

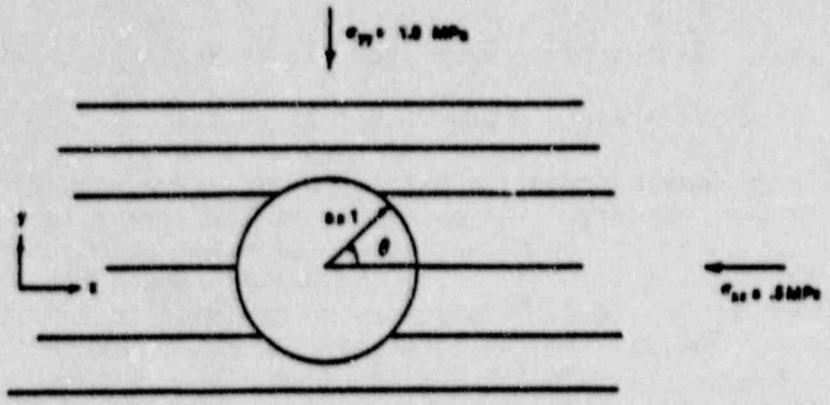


Fig. 7-16 Geometry for Tunnel in a Transversely-Isotropic Rock Mass

One-quarter symmetry is again assumed by fixing the x-velocity along line i=1 and the y-velocity along line j=1. To provide the boundary loading, we use the **APPLY** command on the opposite boundaries:

```
flac> apply press=1.0e6 i=1,21 j=21

flac> apply press=.5e6 i=21 j=21,1
```

These commands apply a biaxial compressive stress to the boundary. The direction of the stress is controlled by the numbering of the nodes across which it is applied. (See the **APPLY** command for greater detail.) We also use the command **init sxx=-.5e6 syy=-1e6** to establish the initial stress state so that equilibrium is accomplished within 1-to-3 timesteps. The stresses can be examined at any time by typing **print sxx syy**.

Once the stresses have come to equilibrium, change the material model and give new properties by using the following commands:

```
flac> model anis (sets all zones to anisotropic
          model)

flac> prop e=.1103e6  xm=.976e6  ym=.953e6

flac> prop nuy=.167  nuz=.165  dens=2000
```

Now, zero the displacements and excavate the tunnel, as before. The problem can now be timestepped to equilibrium using the **STEP** or **SOLVE** command. Approximately 1000 timesteps will be required at about 30 seconds per step on the standard IBM PC.

Plots of the resulting principal stresses and displacement vectors superimposed on the boundary outline are given in Figs. 7-17 and 7-18, respectively. These plots were produced by using the following command sequence.

```
flac> wind -.5,2. -.5,2 (sets a small  window)

flac> set cga pal=0 (sets color screen, palette
          of 0)

flac> plot pen str boun green (plots principal
          stress)

flac> plot pen di max=4 red bo gre (plots dis
          vectors)
```

The keyword PEN is removed if only a screen plot is desired. If you have a two-pen plotter, assign colors in the command line to change the pen color when each new plot is encountered. Compressive principal stresses will always be plotted in color #1, while tension will always be plotted in color #2.

Figure 7-19 plots the calculated-versus-analytical displacements as a function of angle from the x-axis. The maximum error for this example is approximately 10%, and results primarily from the close proximity of the stress boundaries and the coarse grid.

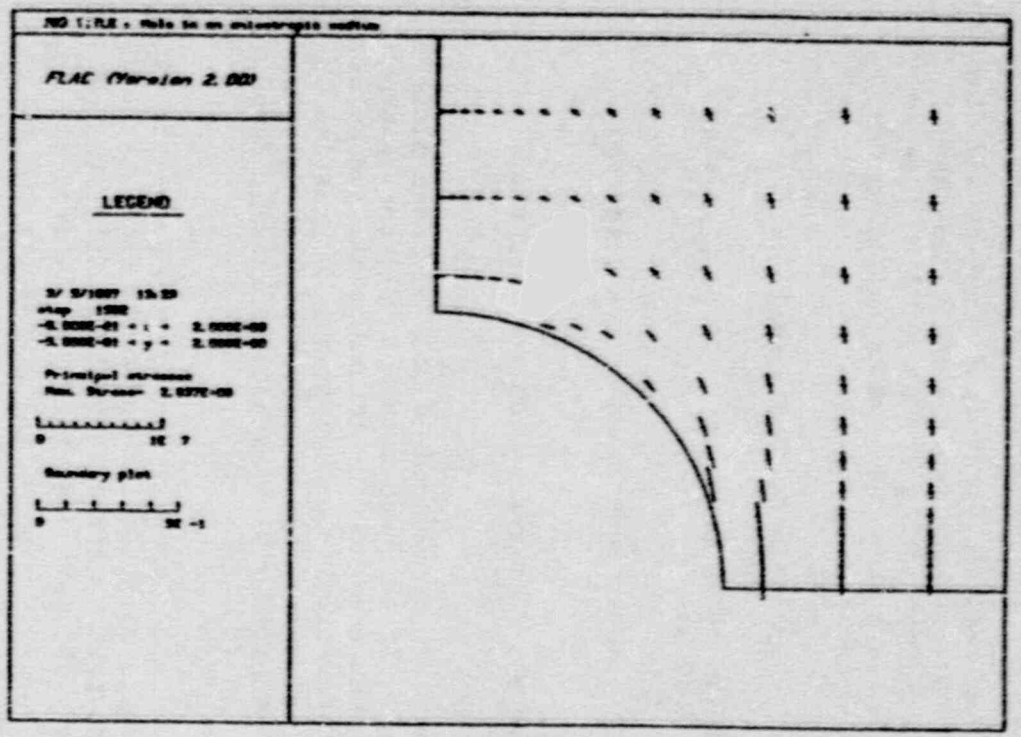


Fig. 7-17 Principal Stress Vectors for a Tunnel in a Transversely-Isotropic Rock Mass

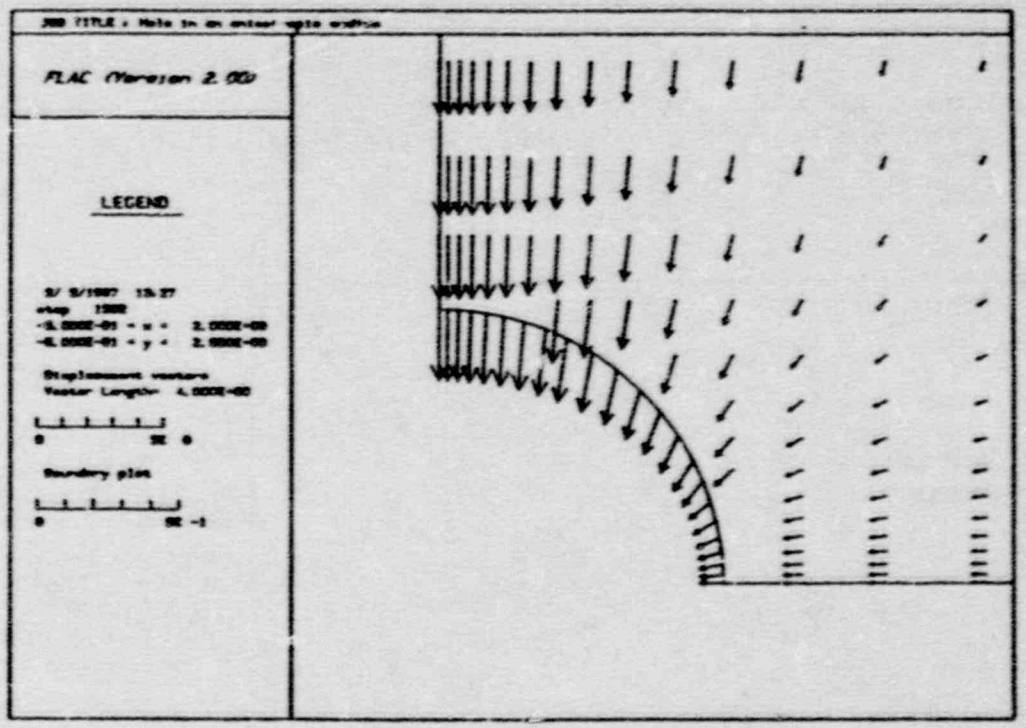


Fig. 7-18 Gridpoint Displacements for the Transversely-Isotropic Example

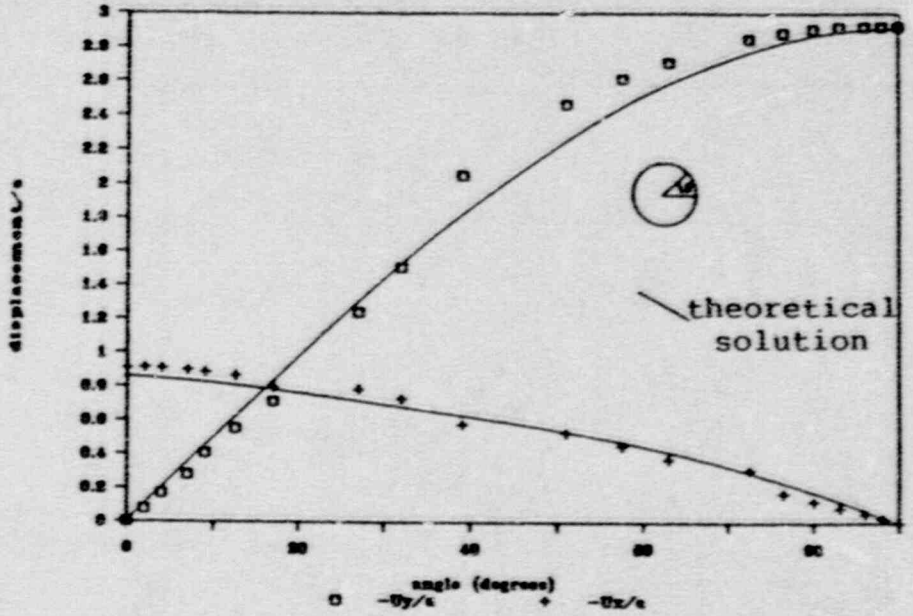


Fig. 7-19 Theoretical vs Predicted X- and Y-Displacements Around Tunnel Boundary

References

Bray, J. "A Study of Jointed and Fractured Rocks —Part II," Felsmechanik und Ingenieungeologie, V(4), 7 (1967).

Eissa, Esa. Stress Analysis of Underground Excavations in Isotropic and Stratified Rock Using the Boundary Element Method, Ph.D. Thesis, Imperial College of Science and Technology, 1980.

Sloan, S. W., and M. F. Randolph. "Numerical Prediction of Collapse Loads Using Finite Element Methods," Int. J. Num. & Analy. Methods in Geomech., 6, 47-76 (1982).

8.0 IMPORTANT POINTS OR CAUTIONS TO NOTE WHEN RUNNING FLAC

This section details some points in running FLAC which we have found may provide pitfalls for some users. As we continue to obtain response from users, this section will be expanded and updated.

8.1 Initializing Variables

It is common practice to initialize the displacements of the grid points between runs to aid in the interpretation of a simulation in which many different excavation stages are used. This can be done because the code does not require the displacements in the calculation sequence—they are determined from the velocities of the grid points as a convenience to the user.

Initialization of the velocities, however, is a different matter. If the velocities of grid points are set to a constant value, they will continue to have this value until set otherwise. Therefore, do not initialize the velocities of the grid to zero simply to clear them—this will affect the simulation results.

8.2 Changing Material Models

FLAC has a limit on the number of different material models you may use during a simulation. The code has been dimensioned to allow the user to have a different material for each zone (if you wish) for the maximum size grid for your version of FLAC. Problems in memory may arise, however,

if you change the material model of the entire grid for large-sized problems.

8.3 Running Problems with In-Situ Field Stresses and Gravity

There are a number of problems in which in-situ field stresses and gravity must be applied to the model. An example of such a problem is deep cut-and-fill mining in which the rock mass is subjected to high in-situ stress fields (i.e., gravity stresses for the limited mesh size can be ignored) but in which the emplaced backfill pillars will develop gravitational stresses which could collapse under the load. The important point to note in these simulations (as in any simulation in which gravity is applied) is that at least two points on the grid must be fixed in space—otherwise, the entire grid will translate due to gravity. If you ever notice the entire grid translating in the negative y-direction, it is a good bet that you have forgotten to fix it in space.

If you desire to have only in-situ stresses varying with depth, use the **INITIAL** command with a depth variation option:

```
flac> ini  sxx=-30e6  (var=0, 10e6)
```

This will cause the xx-stress to vary from -30e6 at the bottom of the grid (i=1, j=1) to -20e6 at the top of the grid (i=n, j=n). Because the i,j range is not give, FLAC assumes the entire grid. The x-variance is set to zero in this example.

F L A C

A P P E N D I X A

DESCRIPTION OF CONSTITUTIVE MODELS

A1. INTRODUCTION

There are six basic consititutive laws provided in FLAC Version 2.0. These are:

(1) elastic, isotropic;
(2) Mohr-Coulomb plasticity;
(3) null;
(4) elastic, transversely isotropic;
(5) ubiquitous joint; and
(6) strain-softening

A short discussion of the theoretical background for each model is given here.

A2. ELASTIC, ISOTROPIC MODEL

The relation of stress to strain in incremental form is expressed by Hooke's law in plane strain as:

$$\begin{aligned} \Delta\sigma_{11} &= \alpha_1 \Delta e_{11} + \alpha_2 \Delta e_{22} \\ \Delta\sigma_{22} &= \alpha_2 \Delta e_{11} + \alpha_1 \Delta e_{22} \\ \Delta\sigma_{12} &= 2G \Delta e_{12} \\ \Delta\sigma_{21} &= \Delta\sigma_{12} \end{aligned} \qquad \text{(A1)}$$

where $\alpha_1 = K + (4/3)G$

$\alpha_2 = K - (2/3)G$

K = bulk modulus, and

G = shear modulus.

$$\Delta e_{ij} = \frac{1}{2}\left[\frac{\partial \dot{u}_i}{\partial x_j} + \frac{\partial \dot{u}_j}{\partial x_i}\right]\Delta t$$

where Δe_{ij} = the incremental strain tensor,

$\dot{u}_i$ = the displacement rate, and

Δt = time step.

In plane stress, these equations become

$$\begin{aligned}
\Delta\sigma_{11} &= \beta_1\,\Delta e_{11} + \beta_2\,\Delta e_{22} \\
\Delta\sigma_{22} &= \beta_2\,\Delta e_{11} + \beta_1\,\Delta e_{22} \\
\Delta\sigma_{12} &= 2G\,\Delta e_{12} \\
\Delta\sigma_{21} &= \Delta\sigma_{12}
\end{aligned} \qquad \text{(A2)}$$

where $\beta_1 = \alpha_1 - (\alpha_2^2/\alpha_1)$ and

$\beta_2 = \alpha_2 - (\alpha_2^2/\alpha_1)$

A3. MOHR-COULOMB PLASTICITY

The plasticity formulation in FLAC assumes an elastic, perfectly plastic solid in plane strain which conforms to a Mohr-Coulomb yield condition and non-associated flow rule.

The yield surface is given by:

$$f = \sigma_1 - N_\phi \sigma_2 + 2C\,(N_\phi)^{1/2} \qquad \text{(A3)}$$

and the plastic potential function is given by

$$g = \sigma_1 - N_\psi \sigma_2 + 2C\,(N_\psi)^{1/2} \qquad \text{(A4)}$$

where $N_\xi = (1 + \sin \xi)/(1 - \sin \xi)$ $\quad [\xi = \phi \text{ or } \psi]$

C = cohesion (positive sign)

ϕ = friction angle

ψ = dilation angle

σ_1 = major principal stress

σ_2 = minor principal stress

The strain increments are assumed to be composed of elastic and plastic parts:

$$\Delta e_1 = \Delta e_1^e + \Delta e_1^p \qquad \text{(A5)}$$

$$\Delta e_2 = \Delta e_2^e + \Delta e_2^p$$

The plastic strain rates are given by the non-associated flow rule:

$$\dot{e}_1^p = \lambda \frac{\partial g}{\partial \sigma_1} = \lambda$$

$$\dot{e}_2^p = \frac{\lambda \partial g}{\partial \sigma_2} = -\lambda N_\psi \qquad \text{(A6)}$$

where λ is the multiplier which must be determined from the stress state.

Multiplying by Δt, Eq. (A6) becomes

$$\Delta e_1^p = \lambda \Delta t$$

$$\Delta e_2^p = -\lambda N_\psi \Delta t \qquad \text{(A7)}$$

The incremental principal elastic stresses are given by:

$$\Delta\sigma_1 = (K + \frac{4}{3} G)\, \Delta e_1^e + (K - \frac{2}{3} G)\, \Delta e_2^e$$

$$\Delta\sigma_2 = (K - \frac{2}{3} G)\, \Delta e_1^e + (K + \frac{4}{3} G)\, \Delta e_2^e \qquad (A8)$$

Substituting Eqs. (A5) and (A7) into Eq. (A8),

$$\Delta\sigma_1 = (K+\frac{4}{3}G)\, (\Delta e_1 - \lambda\Delta t) + (K - \frac{2}{3}G)\, (\Delta e_2 + \lambda N_\psi\, \Delta t)$$

$$\Delta\sigma_2 = (K - \frac{2}{3} G)\, (\Delta e_1 - \lambda\Delta t) + (K+\frac{4}{3}G)\, (\Delta e_2+\lambda N_\psi \Delta t) \qquad (A9)$$

Determining the initial elastic principal stresses by σ_1^I and σ_2^I, the corrected (for plasticity) stresses σ_1, σ_2 are given by

$$\sigma_i^I - \sigma_i = \Delta\sigma_i^I - \Delta\sigma_i \qquad \text{(A10)}$$

where i = 1 or 2.

The initial principal stress increments are given by

$$\Delta\sigma_1^I = (K + \frac{4}{3} G)\, \Delta e_1 + (K - \frac{2}{3} G)\, \Delta e_2$$

$$\Delta\sigma_2^I = (K - \frac{2}{3} G)\, \Delta e_1 + (K + \frac{4}{3} G)\, \Delta e_2 \qquad \text{(A11)}$$

By substituting Eqs. (A9) and (A11) into Eq. (A10),

$$\sigma_1 = \sigma_1^I - \lambda\Delta t\, [(K + \frac{4}{3} G) - (K - \frac{2}{3} G)\, N_\psi]$$

$$\sigma_2 = \sigma_2^I - \lambda\Delta t\, [(K - \frac{2}{3} G) - (K + \frac{4}{3} G)\, N_\psi] \qquad \text{(A12)}$$

or, simplifying,

$$\sigma_1 = \sigma_1^I - \lambda\Delta t\,(\alpha_1 - \alpha_2 N_\psi)$$

$$\sigma_2 = \sigma_2^I - \lambda\Delta t\,(\alpha_2 - \alpha_1 N_\psi) \qquad \text{(A13)}$$

Equation (A13) gives the new principal stresses in terms of the trial initial stress, the material constants, and the multiplier, λ. The λ value can be found since σ_1 and σ_2 must lie on the yield surface (if a non-admissable stress state is detected). This is done by substituting Eq. (A13) into the equation for the yield surface [Eq. (A3)] and equating to zero:

$$\lambda\Delta t = \frac{\sigma_1^I - N_\phi\,\sigma_2^I - 2C\,N_\phi}{\alpha_2(1+N_\phi N_\psi) - \alpha_1(N_\phi + N_\psi)}$$

$$= \frac{f\,(\sigma_1^I,\ \sigma_2^I,\ C,\ N_\phi)}{\gamma} \qquad \text{(A14)}$$

where $\gamma = \alpha_2(1 + N_\phi N\psi) - \alpha_1(N_\phi + N_\psi)$.

Substituting Eq. (A14) into Eq. (A13) yields the corrected principal stress components:

$$\sigma_1 = \overset{I}{\sigma_1} \cdot (\alpha_1 - \alpha_2 N_\psi) \frac{f(\overset{I}{\sigma_1}, \overset{I}{\sigma_2}, C, N_\phi)}{Y} \qquad (A15)$$

$$\sigma_2 = \overset{I}{\sigma_2} - (\alpha_2 - \alpha_1 N_\psi) \frac{f(\overset{I}{\sigma_1}, \overset{I}{\sigma_2}, C, N_\phi)}{Y}$$

The program first determines the σ_{xx}, σ_{yy} and σ_{xy} elastic stress increments and converts them to principal stresses. These components are then substituted into the equation for the yield surface [Eq. (A3)]. If,

$f<0$, then the direction cosines of the principal stresses are found and the plastic corrections are made [Eq. (A15)] to require conformity to the yield surface.

$f>0$, then no corrections are necessary.

The principal stresses are then resolved back to the x,y coordinate plane by the direction cosines determined previously.

A4. NULL MODEL

The null model simply represents an excavated zone.

A5. ANISOTROPIC ELASTICITY

For a transversely isotropic body, FLAC assumes that the plane of isotropy lies within the x-z plane (Fig. A1).

For a general orthotropic elastic body, the stress-strain equations are given by (Lekhnitskii, 1981):

$$\begin{aligned}
\Delta e_{11} &= S_{11}\Delta\sigma_{11} + S_{12}\Delta\sigma_{22} + S_{13}\Delta\sigma_{33} \\
\Delta e_{22} &= S_{21}\Delta\sigma_{11} + S_{22}\Delta\sigma_{22} + S_{23}\Delta\sigma_{33} \\
\Delta e_{33} &= S_{31}\Delta\sigma_{11} + S_{32}\Delta\sigma_{22} + S_{33}\Delta\sigma_{33} \qquad \text{(A-16)} \\
\Delta e_{23} &= (1/2)S_{44}\Delta\sigma_{23} \\
\Delta e_{13} &= (1/2)S_{55}\Delta\sigma_{13} \\
\Delta e_{12} &= (1/2)S_{66}\Delta\sigma_{12}
\end{aligned}$$

Where:

$$S_{11} = \cos^4\phi/E_1 + (1/G_{12} - 2\nu_{12}/E_1)\sin^2\phi\,\cos^2\phi + \sin^4\phi/E_2$$

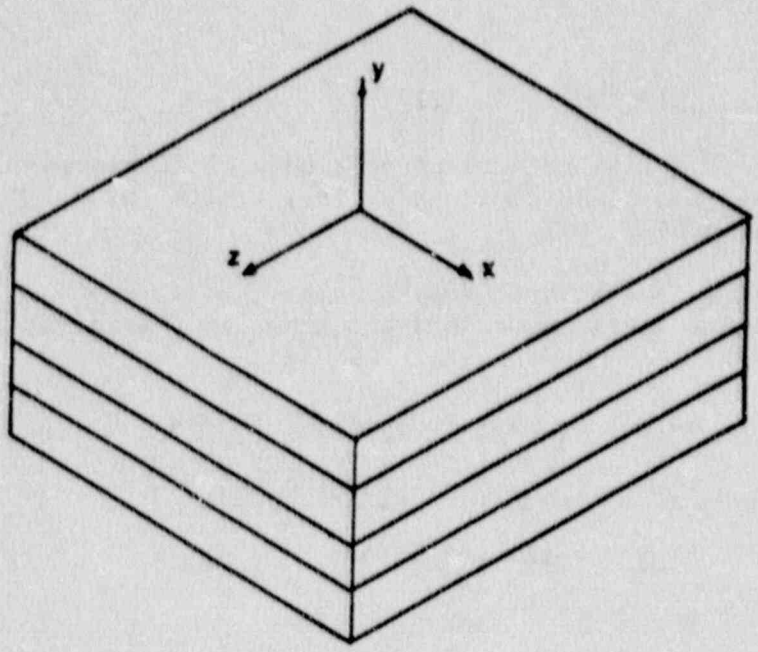


Fig. A-1 Transverse Isotropy Coordinate Axes Convention (x-z direction is plane of isotropy)

$$S_{22} = \sin^4\phi/E_1 + (1/G_{12} - 2\nu_{12}/E_1)\sin^2\phi\ \cos^2\phi + \cos^4\phi/E_2$$

$$S_{12} = (1/E_1 + 1/E_2 + 2\nu_{12}/E_1 - 1/G_{12})\sin^2\phi\cos^2\phi - \nu_{12}/E_1$$

$$S_{13} = -[(\nu_{23}/E_2)\sin^2\phi + (\nu_{13}/E_1)\cos^2\phi]$$

$$S_{23} = -[(\nu_{23}/E_2)\cos^2\phi + (\nu_{13}/E_1)\cos^2\phi]$$

$$S_{33} = 1/E_3$$

$$S_{44} = (\cos^2\phi/G_{23}) + (\sin^2\phi/G_{13})$$

$$S_{55} = (\sin^2\phi/G_{23})/(\cos^2\phi/G_{13})$$

$$S_{66} = 4(1/E_1 + 1/E_2 + 2\nu_{12}/E_1 - 1/G_{12})\sin^2\phi\cos^2\phi + 1/G_{12}$$

ϕ = angle of anisotropy anti-clockwise from the x-axis (Fig. A-2).

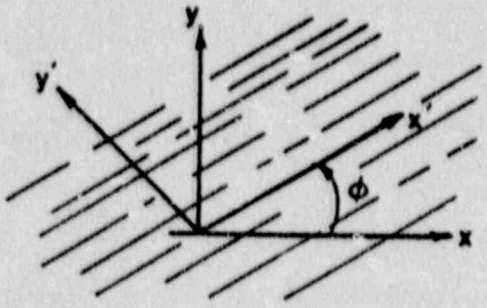


Fig. A-2 Planes of Elastic Anisotropy Oriented at an Angle ϕ From the x-Axis

A state of plane stress with respect to the x-y plane is obtained by setting $\Delta\sigma_{33}=\Delta\sigma_{13}=\Delta\sigma_{23} = 0$ in Eqs. (A16). This gives:

$$\Delta e_{11} = S_{11}\Delta\sigma_{11} + S_{12}\Delta\sigma_{22}$$

$$\Delta e_{22} = S_{12}\Delta\sigma_{11} + S_{22}\Delta\sigma_{22}$$

$$\Delta e_{12} = 1/2\ S_{66}\Delta\sigma_{12}$$

The inverse form of these equations are:

$$\Delta\sigma_{11} = (S_{22}\Delta e_{11} - S_{12}\Delta e_{22})/(S_{11}S_{22} - S^2_{12})$$

$$\Delta\sigma_{22} = (-S_{12}\Delta e_{11} + S_{11}\Delta e_{22})/(S_{11}S_{22} - S^2_{12})$$

$$\Delta\sigma_{12} = 2\Delta e_{12}/S_{66}$$

For a state of plane strain in the x-y plane is obtained from Eqs. (A16) by setting

$$\Delta e_{33} = \Delta e_{13} = \Delta e_{23} = 0$$

This results in:

$$\Delta\sigma_{13} = \Delta\sigma_{23} = 0$$

$$\Delta\sigma_{33} = -(S_{13}\Delta\sigma_{11} + S_{23}\Delta\sigma_{22})/S_{33}, \text{ and,}$$

$$\Delta e_{11} = (S_{11} - S^2_{13}/S_{33})\Delta\sigma_{11} + (S_{12} - S_{13}S_{23}/S_{33})\Delta\sigma_{22}$$

$$\Delta e_{22} = (S_{12} - S_{13}S_{23}/S_{33})\Delta\sigma_{11} + (S_{22} - S^2_{23}/S_{33})\Delta\sigma_{22}$$

$$\Delta e_{12} = 1/2\ S_{66}\Delta\sigma_{12}$$

The inverse form of these equations is:

$$\Delta\sigma_{11} = C_{11}\Delta e_{11} + C_{12}\Delta e_{22}$$

$$\Delta\sigma_{22} = C_{22}\Delta e_{11} + C_{22}\Delta e_{22}$$

$$\Delta\sigma_{12} = 2\, C_{66}\Delta e_{12}$$

Where:

$$C_{11} = (S_{22} - S^2_{23}/S_{33})So^2$$

$$C_{12} = -(S_{12} - S_{13}S_{23}/S_{33})/So^2$$

$$C_{22} = (S_{11} - S_{13}^2/S_{33})/So^2$$

$$C_{66} = 1/S_{66}$$

$$So^2 = S_{11}S_{22} - S_{12}^2 + (2S_{12}S_{13}S_{23} - S_{11}S_{23}^2 - S_{22}S_{13}^2)/S_{33}$$

For the case of transverse isotropy with the plane of isotropy in the x-z plane,

$$E_1 = E_3$$

$$\nu_{12} = \nu_{32}$$

A6. UBIQUITOUS JOINT MODEL

The ubiquitous joint model is an anisotropic plasticity model which assumes a series of weak planes embedded in a Mohr-Coulomb solid. Yield may occur in either the solid or along the slip plane, or both, depending on the material properties of the solid and plane, the stress state, and the angle of the slip planes.

Figure A-3 illustrates the weak plane existing in a Mohr-Coulomb solid and the global (x-y) and local (x'-y') coordinate frame.

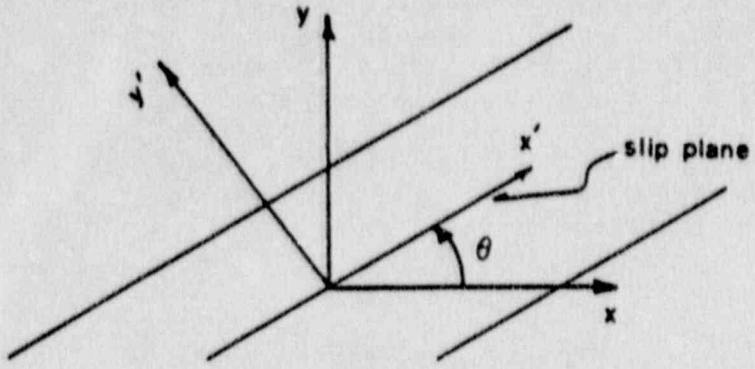


Fig. A-3 A Slip Plane Oriented at an Angle θ to the Global Reference Frame

The global stresses must be resolved parallel and perpendicular to the potential slip surface:

$$\sigma'_{11} = \sigma_{11} \cos^2\theta + 2\sigma_{12} \sin\theta \cos\theta + \sigma_{22} \sin^2\theta$$

$$\sigma'_{22} = \sigma_{11} \sin^2\theta - 2\sigma_{12} \sin\theta \cos\theta + \sigma_{22} \cos^2\theta \qquad \text{(A17)}$$

$$\sigma'_{12} = -(\sigma_{11} - \sigma_{22}) \sin\theta \cos\theta + \sigma_{12}(\cos^2\theta - \sin^2\theta)$$

where θ = joint angle counterclockwise from the x global axis.

The initial stress increments above (σ_{ij}) are found from the plasticity model presented in Section A2. If the matrix material is currently elastic, then σ_{ij} will simply be the elastic stress increments. If, however, the matrix is yielding, then σ_{ij} will be the stress increments corrected for plasticity. The stresses along the joint are examined for yield:

$$|\sigma'_{12}| < |\sigma'_{22} \tan\phi| + C \qquad \text{(A18)}$$

where ϕ = joint friction angle, and

C = joint cohesion (positive sign).

If Eq. (A18) is true, then no corrections to the stresses are necessary because slip is not taking place. If Eq. (A18) is false, then slip is occuring, which requires stress corrections. The "excess" shearing stress along the joint is given by

$$\Delta\sigma'_{12} = \text{sign}\,(|\sigma'_{12}|, \sigma'_{12}) - \sigma'_{12} \qquad \text{(A19)}$$

The corrections to the global stresses can be computed as follows by using the reverse stress transformation:

$$\Delta\sigma_{11} = -2\Delta\sigma'_{12} \cos\theta \sin\theta$$

$$\Delta\sigma_{22} = 2\Delta\sigma'_{12} \cos\theta \sin\theta$$

$$\Delta\sigma_{12} = \Delta\sigma'_{12} (\cos^2\theta - \sin^2\theta)$$

assuming that $\Delta\sigma'_{22} = \Delta\sigma'_{11} = 0$ (i.e., no joint dilation).

These stress corrections are added into the trial stresses to be used in the next timestep.

A7. STRAIN-SOFTENING/HARDENING MODEL

This constitutive model allows the user to represent arbitrary non-linear material hardening and softening behavior based on the variation of the cohesion, friction, and dilatancy with plastic strain. The model is based on Mohr-Coulomb elasto-plasticity with non-associated flow rule, as described earlier. The difference, however, lies in the ability of the cohesion, friction and dilation to harden or soften after the onset of plastic yield. In the previous Mohr-Coulomb plasticity model, the cohesion, friction and dilation are assumed to remain constant. Here, the user defines the cohesion, friction and dilation as piecewise linear functions of the plastic strain. The code determines the total plastic strain at each time increment and causes the cohesion, friction and dilation to conform to the user-defined functions.

Determination of Plastic Strain Increments

Consider the Mohr-Coulomb yield surface in stress space defined by the mean and deviatoric stress components p and q, where

$$p = \frac{1}{2} \left(\sigma_{11} + \sigma_{22} \right)$$

$$q = \frac{1}{2} \left[\left(\sigma_{11} - \sigma_{22} \right)^2 + 4\sigma_{12}^2 \right]^{1/2} \qquad \text{(A20)}$$

The Mohr-Coulomb yield surface is given by (Fig. A-4):

$$f = (p - q) - (p + q)N\phi + 2C(N\phi)^{1/2} \qquad (A21)$$

where $N\phi = \dfrac{1 + \sin\phi}{1 - \sin\phi}$,

ϕ = angle of internal friction, and

C = cohesion.

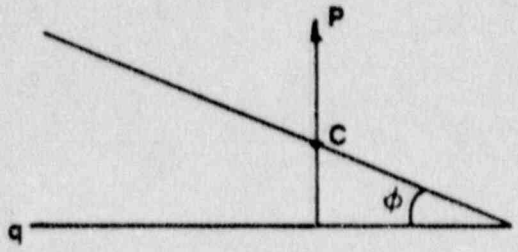


Fig. A-4 Mohr-Coulomb Yield Surface in p-q Space, Compression Negative

The plastic potential function is given by:

$$g = (p - q) - (p + q)N\psi + 2C(N\psi)^{1/2} \qquad \text{(A22)}$$

where $N\psi = \dfrac{1 - \sin\psi}{1 + \sin\psi}$, and ψ = dilation angle.

The plastic strain increments are given in terms of the plastic potential and a multiplier:

$$\Delta e^{P}_{11} = - \lambda \frac{\partial g}{\partial \sigma_{11}}$$

$$\Delta e^{P}_{22} = - \lambda \frac{\partial g}{\partial \sigma_{22}} \qquad \text{(A23)}$$

$$\Delta e^{P}_{12} = -\lambda \frac{\partial g}{\partial \sigma_{12}}$$

Using the chain rule, the above derivations may be expanded:

$$\frac{\partial g}{\partial \sigma_{11}} = \frac{\partial g}{\partial p} \cdot \frac{\partial p}{\partial \sigma_{11}} + \frac{\partial g}{\partial q} \cdot \frac{\partial q}{\partial \sigma_{11}}$$

$$\frac{\partial g}{\partial \sigma_{22}} = \frac{\partial g}{\partial p} \cdot \frac{\partial p}{\partial \sigma_{22}} + \frac{\partial g}{\partial q} \cdot \frac{\partial q}{\partial \sigma_{22}} \qquad \text{(A24)}$$

$$\frac{\partial g}{\partial \sigma_{12}} = \frac{\partial g}{\partial p} \cdot \frac{\partial p}{\partial \sigma_{12}} + \frac{\partial g}{\partial q} \cdot \frac{\partial q}{\partial \sigma_{12}}$$

and

$$\frac{\partial g}{\partial p} = (1 - N\psi)$$

$$\frac{\partial g}{\partial q} = -(1 - N\psi) \qquad \text{(A25)}$$

$$\frac{\partial p}{\partial \sigma_{11}} = \frac{1}{2}; \quad \frac{\partial p}{\partial \sigma_{22}} = \frac{1}{2}; \quad \frac{\partial p}{\partial \sigma_{12}} = 0$$

$$\frac{\partial q}{\partial \sigma_{11}} = \frac{(\sigma_{11} - \sigma_{22})}{4q}; \quad \frac{\partial q}{\partial \sigma_{22}} = -\frac{(\sigma_{11} - \sigma_{22})}{4q}; \quad \frac{\partial q}{\partial \sigma_{12}} = \frac{\sigma_{12}}{q}$$

Substitution of Eqs. (A24 and A25) into Eq. (A23) yields

$$\Delta e^{p}_{11} = -\lambda \left[\frac{1}{2} (1 - N\psi) - (1 + N\psi) \frac{(\sigma_{11} - \sigma_{22})}{4q} \right]$$

$$\Delta e^{p}_{22} = -\lambda \left[\frac{1}{2} (1 - N\psi) + (1 + N\psi) \frac{(\sigma_{11} - \sigma_{22})}{4q} \right]$$

$$\Delta e^{p}_{12} = -\lambda \left[(1 + N\psi) \frac{\sigma_{12}}{q} \right]$$

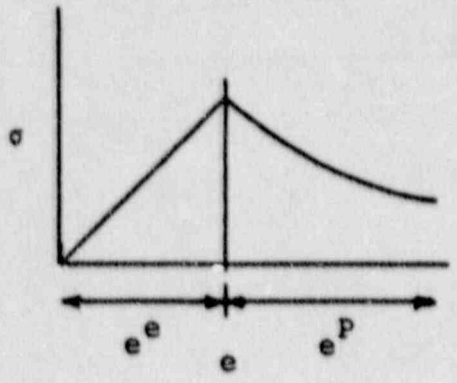
The value of λ must next be determined to define the plastic strain increments. An interactive technique which places the stress state on the yield surface (Newton's method) is used to determine λ. Once λ is known, the plastic strain increments are determined.

User-Defined Functions for Cohesion, Friction and Dilation

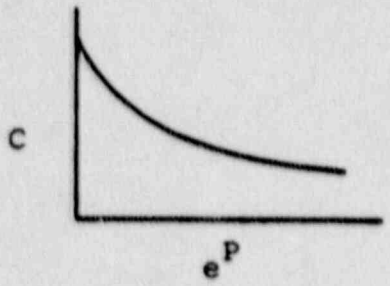
The user defines the cohesion, friction and dilation to be functions of the total plastic strain. Once plastic yield begins, the code keeps track of the total plastic strain and determines the cur-

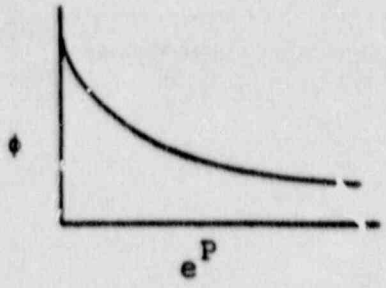
rent value of cohesion, friction and dilation which corresponds to this strain. In the next timestep, these new values are used in the calculation of stress and strain.

The cohesion, friction and dilation are defined as piecewise linear segments of a generally non-linear function of the total plastic strain. For example, assume the rock mass has a stress-strain curve which softens upon yield and attains some residual strength:

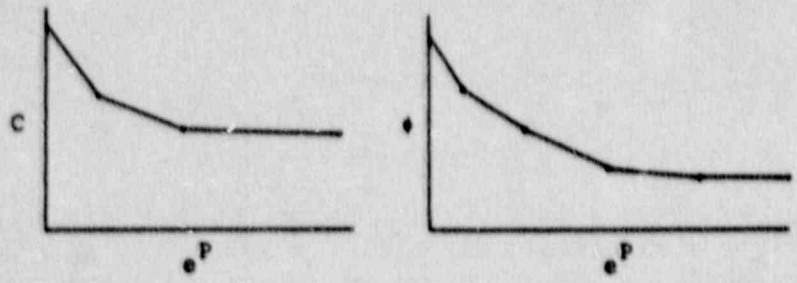


The stress-strain curve is linear to the point of yield; therefore, the strain will be elastic only (e^e). After yield, the total strain will be composed of elastic and plastic parts. This model requires the user to define the cohesion, friction and dilation variance as a function of the plastic portion of the total strain. In reality, these functions are probably non-linear:





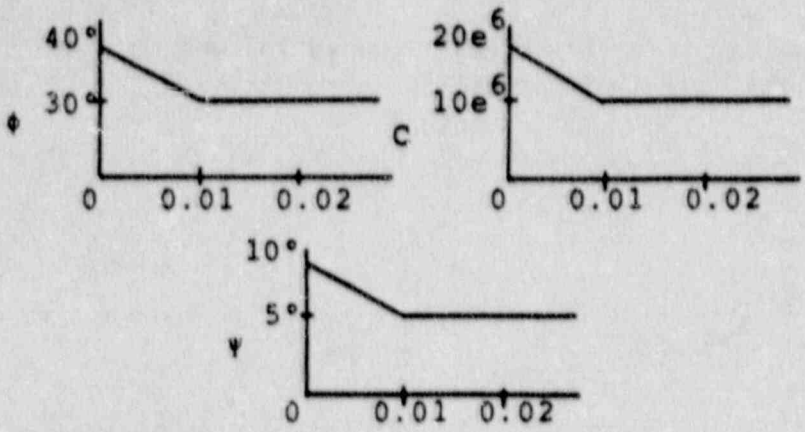
However, they may be approximated for use in FLAC as a set of linear segments:



These functions are input to the code via the **PROP** and **TABLE** commands. For example:

```
model ss
prop s=11.5e9 b=8.62e9 d=2000 ftab=1 ctab=2 dtab=3
prop fric=40 coh=20e6 dil=10
table 1  0,40  .01,30
table 2  0,20e6, .01,7e6
table 3  0,10  .01,5
```

Here, the friction function is defined in table 1, the cohesion in table 2, and the dilation in table 3. Note that the initial friction, cohesion and dilation must be defined (here, to be 40°, 20 MPa and 10°, respectively). The functions each consist of two linear segments, as shown in the following figures:



Hardening behavior can be produced by an increase in these parameters with increasing plastic strain. This model is experimental in nature and given to the user primarily for research purposes.

REFERENCE

Lekhnitskii, S. G. Theory of Elasticity of an Anisotropic Body. Moscow: Mir Publishers, 1981.

F L A C

A P P E N D I X B

DETERMINING EQUILIBRIUM CONDITIONS WHEN USING FLAC

When using an explicit code such as FLAC, it is important that the user identify when the simulation has arrived at an equilibrium state. FLAC is a dynamic code which solves static problems by properly damping the response of the structure. The final static equilibrium solution for a dynamic problem can be illustrated by the standard problem of damped vibrations of a mass on a spring (Fig. B-1). The system response is governed by the following linear differential equation:

$$my'' + cy' + ky = 0$$

where m = mass

c = damping constant

k = spring stiffness

y' = mass velocity

y'' = mass acceleration

The damping is simulated by the dashpot beneath the weight. If the weight is displaced and released, several forms of motion are possible. If the system is undamped (assuming no frictional energy dissipation in the spring), the weight will

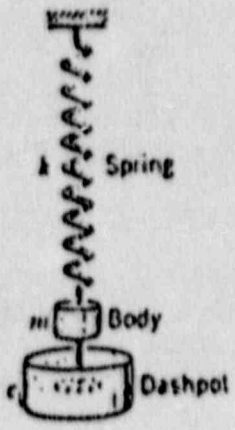


Fig. B-1 Body Subject to Damped Vibrations

move with harmonic oscillations (Fig. B-2). If the mass is damped, three responses are possible: under-damped, overdamped, and critically damped. The first two cases are illustrated in Figs. B-3 and B-4.

In the case of underdamping (c very small), oscillations will occur, but the motion of the mass will decay to the static solution of zero after some period of time, t. When the value of c is large, the response will be overdamped, approaching the static solution of zero without oscillations. The speed with which the static solution is approached depends on the magnitude of the damping constant, c. FLAC automatically determines the damping constants used in the simulation and the response is slightly overdamped to ensure stable convergence to the solution.

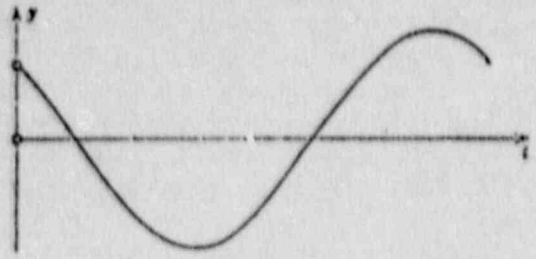


Fig. B-2 Undamped Oscillations

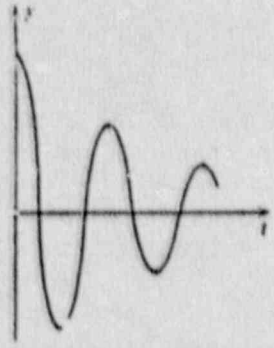


Fig. B-3 Underdamped System (oscillates but approaches equilibrium with time)

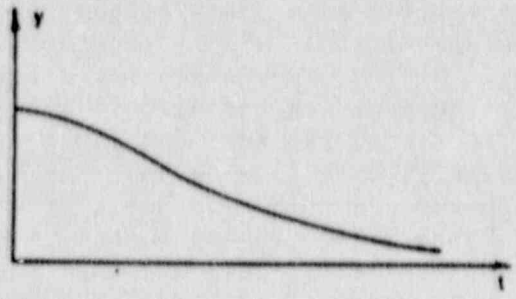


Fig. B-4 Overdamped System (no oscillations)

To minimize the time (both model and real) necessary to arrive at the static solution, the question arises as to when the model sufficiently converges to the static equilibrium solution. For our purposes it is only necessary that we be reasonably "close" to the static equilibrium solution. At equilibrium, the following occur.

(1) The velocities of the grid points will be zero.

(2) The out-of-balance force at each grid point will be zero.

(3) The displacements of each grid point will approach some constant value.

(4) The stresses in each zone will approach some constant value.

Probably the best indicator of equilibrium within FLAC is noted in (2), above. At each timestep, the code determines the forces acting on each grid point. As an illustration, examine the problem of a block with an applied edge pressure and fixed boundaries (Fig. B-5). Initially (timestep 0), grid point 1 will have a y-direction force equivalent to the applied pressure acting on one side in the negative direction. The out-of-balance force will be very large at this time because the effects of the applied pressure have not propagated through the mesh (remember, we are working with an explicit code). Eventually, as additional timesteps occur, the boundary effects will propagate through the body, and the out-of-balance force

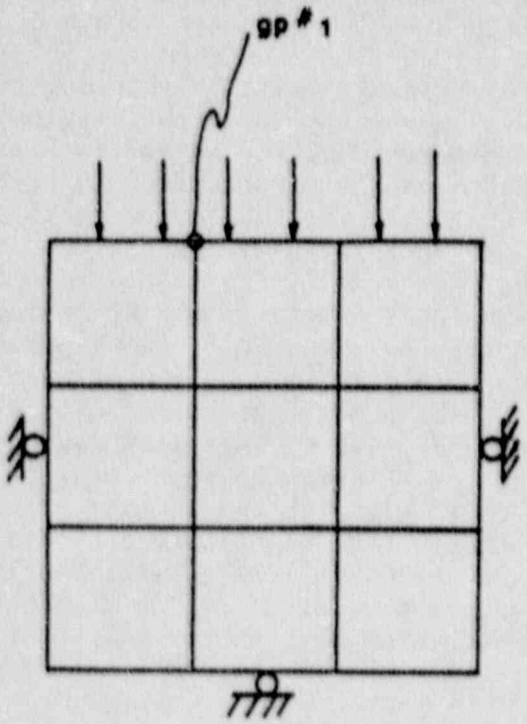


Fig. B-5 A Grid With Edge Pressure

will approach zero at each grid point (Fig. B-6). When these out-of-balance forces are sufficiently small, the static solution has been obtained.

But what is "sufficiently small"? This depends on the problem. For a problem in which in-situ field stresses of 30 MPa are applied, an out-of-balance force of 1 kNewton may be acceptable. In problems with gravitational stresses (which are small), smaller out-of-balance force limits of, say, 100-200 Newtons are required. It is a good idea for the new user to plot the maximum out-of-balance force as a function of timestep for each problem (FLAC gives you this data at the completion of each run); this will allow examination of the speed of convergence. It is also advantageous to choose a grid point at which maximum displacement is likely to occur (e.g., at the surface of an excavation) and to plot its displacement as a function of time step (Fig. B-7). You will see the changes in successive readings approach zero as the model approaches an equilibrium position.

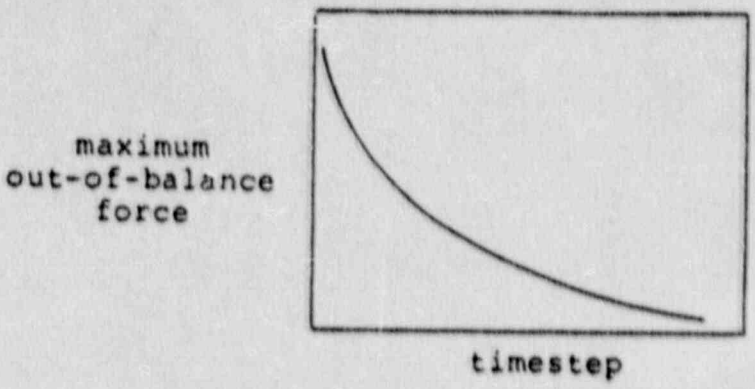


Fig. B-6 Maximum Out-of-Balance Force Will Approach Zero With Increasing Timestep

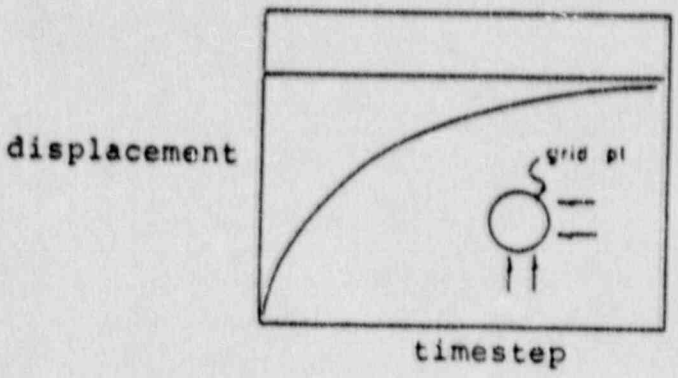


Fig. B-7 Displacement of a Grid Point at an Excavation Will Approach a Constant Value (If Stable) With Increasing Timestep

F L A C

A P P E N D I X C

ERROR AND WARNING MESSAGES

Error Message	Meaning
Command not recognized - type help for list	The typed command is not found in FLAC's command list. Check for spelling error.
Bad integer	The command requires an integer.
Bad number	The command requires a real number. The number has probably been omitted.
Bad keyword	FLAC cannot recognize the keyword typed in. This means either a spelling error or that this keyword does not apply to the primary command given.

Cannot re-define grid at this stage Type "new" to start a new problem.	The grid cannot be re-defined in the middle of a run. Use the **NEW** command to start new run.
End of file	Has come to the end of a call or batch file.
Error reading input	An error has been encountered in reading a call or batch file.
Missing parameter number	A command requires a greater number of parameters than has been entered. The number of the missing parameter is given.
Memory overflow	An excessive number of changes have made to the material types for a problem, resulting in a memory overflow.
Too many zones	The user has attempted to define a grid with more zones than allowed by machine memory.

One of the models given above must be specified	A material model has been specified which does not exist. Recheck the spelling of the model name.
Grid is not yet specified	The grid must be specified using the **GRID** command before issuing any command which relates to the grid.
Pressure must be applied between at least two grid-points on a single row or column	A pressure has been specified at a single gridpoint.
Zero or negative density in one or more zones	An attempt has been made to timestep before assigning a density to the zones.
Cannot open file	A call or batch file does not exist.
Error accessing file	A save file does not exist or there has been an error during accessing of the file.

Saved state created by a different version of FLAC	The version number of the current FLAC program is not the same as that used to create the save file.
Nesting of command files not allowed	An input data file cannot be called from an input data file.
Missing filename	File name for **CALL**, **SAVE** or **RESTORE** not given.
Bad geometry, impossible to continue	A grid has been deformed such that: (a) the area of a quadrilateral is negative; or (b) one member of each pair of triangles does not have an area greater than 20% of the total quadrilateral area. The first quadrilateral zone which has been unacceptably deformed is printed.
Bad i-j range given	The i-j range for a command falls outside the grid.

Bad contour interval	The user has specified a non-sensical contour interval using the **int=** switch.
Table out of range	A table has been requested which does not exist.
The joint friction angle cannot be 90 degrees	The joint friction angle has been incorrectly specified for the ubiquitous joint model.
Bad properties for anisotropic model... please check	Properties have been specified for the anistropic elastic model which do not meet energy conservation requirements.
One or more moduli not specified for anisotropic model	Check to make sure all required properties have been input for the anisotropic model.
Not enough memory for this model	There is not enough remaining RAM for this material model in the present grid.

Cannot plot - wrong model	An attempt has been made to plot a grid model variable for a model which is not in use in the current grid.
Connectivity of null-zone region is bad	The GEN command shape generator has been used to force connected excavated regions to a given shape. This is not allowed.
Line has zero length	The GEN LINE command has attempted to create a line segment of zero length
Missing BEGIN keyword	An attempt has been made to specify a structural element, but the BEGIN keyword has been left out.
Missing END keyword	An attempt has been made to specify a structural element, but the END keyword has been left out.

Properties undefined for at least one element	The properties of at least one structural element have not been defined.
Zero length	A structural element has zero length.
History traces have all been reset	The histories have all been reset. No histories will be kept unless new variables are specified.
History trace # specified does not exist	A history **DUMP**, **WRITE**, or **RESET** command has requested a history number which does not exist.
Error in reading/writing history files	An error has occured in reading/writing of the history files upon saving or restarting. It is likely that this data has been lost in file creation error.

Cannot continue re-zoning; bad geometry	In attempting to re-zone the mesh based on a **GEN** command, a negative or small area has been encountered. Attempt re-zoning with different parameters or in a different order.
Zero or negative radius	A zero or negative radius has been input in the **GEN CIRCLE** or **ARC** commands.
Angle must be positive	The angle for the **GEN ARC** command must be positive.

FLAC

APPENDIX D

Interface Logic

It is often desirable in geomechanics to model planes within or bounding a solid body upon which slip or separation can occur. Some instances, for example, include:

(1) joint or fault planes in a rock mass;

(2) bedding surfaces in a rock mass;

(3) the interface between a foundation and the soil;

(4) contact plane between a bin and the solid it contains; and

(5) contact of two colliding objects.

The FLAC code provides the ability to model interfaces that have behavior characterized by a Mohr-Coulomb criterion. The interfaces may be assigned cohesion, friction, tensile strength, normal and shear stiffness, and may be "glued" to prevent slip or separation. Although there is no restriction on the number of interfaces, it generally is not reasonable to attempt to model more than a few interfaces with the present code. Future releases of the code will allow automatic generation of interfaces but, at present, the required manual generation can be restrictive if many interfaces are generated.

Use of the interface logic requires the following general steps.

1. Prior to developing the grid, carefully define where the interfaces will lie with respect to excavations, boundaries, etc.

2. Define the initial grid, deform to the desired shape using **GEN** and/or **INITIAL** commands.

3. Null out zones to form the faces of the interface. If the interface lies along a line of horizontal or vertical zones, it is only necessary to null a constant i- or j-line. If the interface is to "cut" across zone boundaries, the **GEN LINE** command must be used to form the faces of the interface. The area between these lines can be nulled using the **MOD NULL REG=I,J** command.

4. If necessary, use the **INITIAL** command to move any gridpoints to form the desired faces.

5. Move opposing faces together (not overlapping) using the **GEN** or **INITIAL** commands.

6. Assign properties to the interface, including normal and shear stiffness, cohesion, friction and tensile strength. If properties are not assigned, they are assumed to be zero.

7. Check gridpoint masses to make certain that stiffnesses are reasonable and to ensure that the solution will be numerically stable.

8. Timestep the problem to initial stress equilibrium, making certain that force equilibrium is obtained.

9. Reset the x- and y-displacements to zero (usually), and perform model alterations (e.g., excavation, load application, etc.).

10. Examine slip, displacement along interfaces.

Each of the above steps is examined in more detail here.

First, the interface(s) must be generated. Figure D-1 shows a typical interface and the nomenclature for describing it.

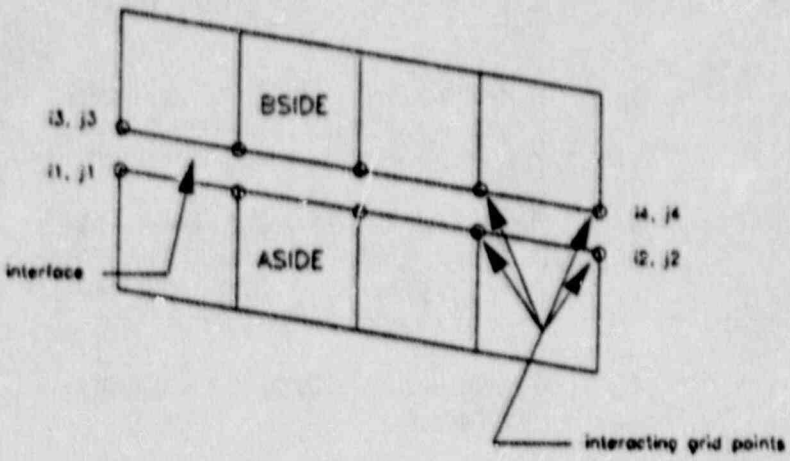


Fig. D-1 Interface Nomenclature

The two interacting faces of the interface are termed the ASIDE and the BSIDE. These faces are created initially by nulling the region between them. The faces may then be moved into contact by using the **GEN** or **INITIAL** commands to create a joint.* This process is illustrated in Fig. D-2 for the creation of horizontal or diagonal interfaces.

*The opposing faces may be left apart, if desired, for special purposes, such as modeling impact.

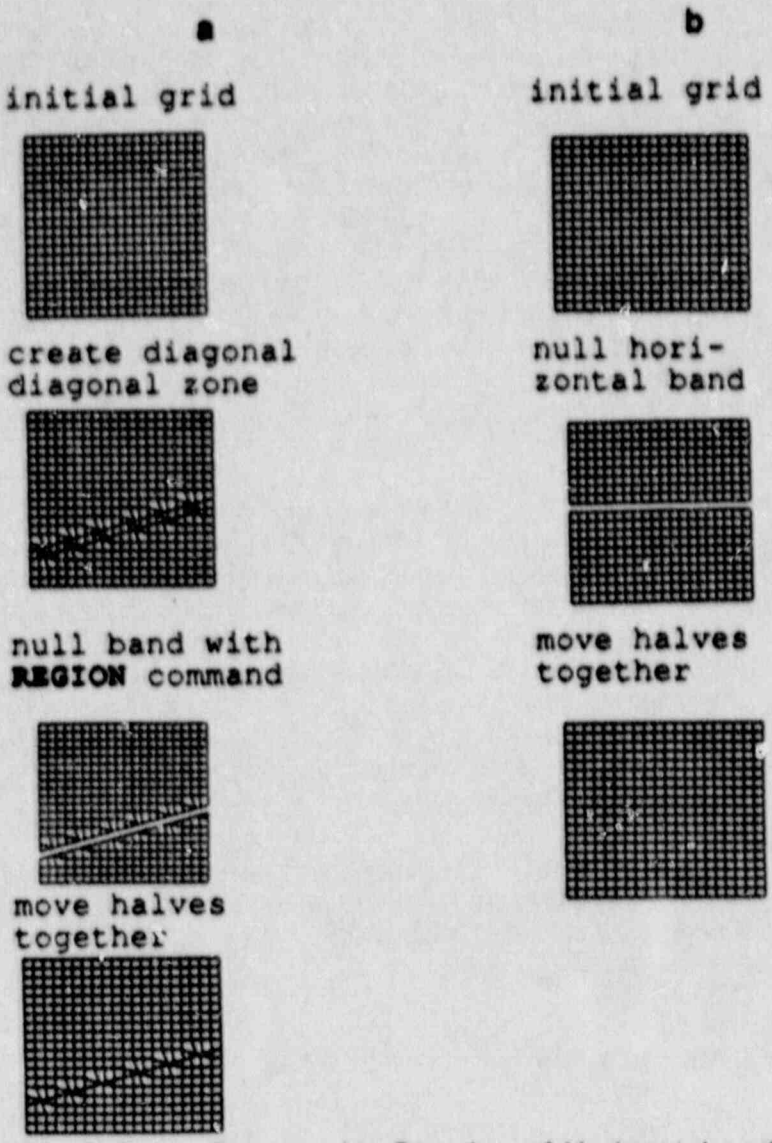


Fig. D-2 Process for Creation of Horizontal and Diagonal Interfaces

The properties of the interface must be set prior to timestepping. The cohesion (Pa, psi, etc.) friction angle (degrees) and tensile strength (Pa, psi, etc.) are self-explanatory, but setting stiffness values requires additional explanation. As stated earlier, the interface can be thought of as a series of normal and shear springs which connect the two solid surfaces which bound it. The stiffness of the normal and shear "springs" determines the normal and shear displacement of the opposing faces. Although FLAC is used to solve static problems, the solution procedure is dynamic in nature, and the stiffnesses need to be set properly to allow a numerically-stable solution.

As a general rule, the normal and shear stiffness are set equal at roughly 10^7-10^9 GPa/m. The values may be somewhat larger, depending on the material type. Once the interface has been formed and properties assigned, the stiffness may be checked by evaluating the values which the code calculates for the masses of each gridpoint. It is desirable that the masses of the gridpoints which lie on the faces of the interface be approximately the same as those surrounding the interface.

We examine a typical problem to illustrate the setting of the interface properties. Consider a block of rock discretized into a 10x10 grid:

```
gr 10,10
m e
prop s=14.4e9 b=19.2e9 d=2000
```

```
* create horizontal interface
gen line 0,3 10,6
gen line 0,5 10,8
*the gen line command creates multiple regions
*in this case which are nulled one at a time
mod null reg=1,4
mod null reg=5,5
mod null reg=7,7
mod null reg=10,7
*move faces together
ini y add -2 reg=1,10
*declare interface
int 1 as from (1,4) to (11,7)/bs from (1,6) to (11,9)
int 1 kn=1e10 ks=1e10 coh=0 fric=5
*fix bottom and sides of grid
fix y j=1
fix x i=1
fix x i=11
*apply gravity
set grav=9.81
*large strain
set large
his nst=5
his ydis i=5 j=11
his xdis i=5 j=11
his unbal
*step to calculate grid point masses
step 300
save grav.sav
free x i=1
free x i=11 j=9,11
ste 1000
save step1.sav
ret
```

Figure D-3 shows the grid as formed with the **INTERFACE** command. Figure D-4 shows the plot of the boundary with displacement vectors as slip occurs.

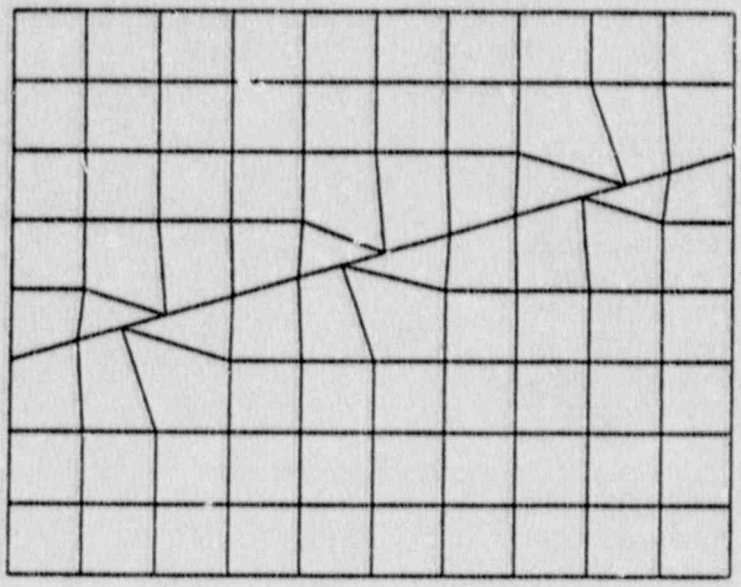


Fig. D-3 Grid Formed by INTERFACE Command

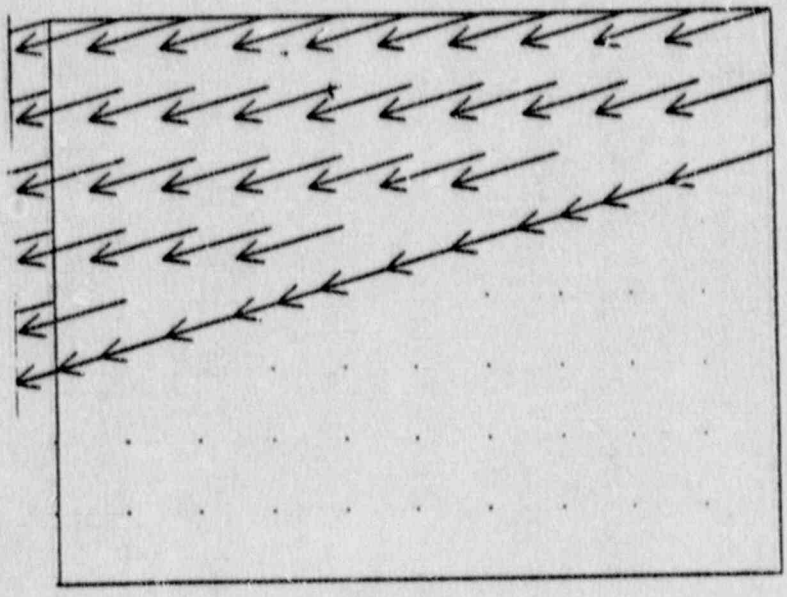


Fig. D-4 Displacement Factors with Interface

The **PRINT INTERFACE** command is used to obtain information on the normal and shear forces and displacements across the interface and the interacting gridpoints and to indicate whether slip has occurred across any element of the interface. The following printout is the result of **PRINT INTERFACE** after 2000 timesteps of slip.

```
Interface  1
-----------
 Friction    cohesion          ks          kn       t-bond  glued?
5.000E+00  0.000E-01  1.000E+10  1.000E+10  0.000E-01    no
                    Side-A
                    ------
  i   j  normal str    shear str    slip?       unit normal              length
  1   4  -9.297E+04   -8.134E+03    yes   2.873E-01  -9.578E-01   2.390E-01
  2   4  -9.079E+04   -7.943E+03    yes   2.873E-01  -9.578E-01   3.113E-01
  3   4  -7.320E+04   -6.405E+03    yes   2.874E-01  -9.578E-01   3.113E-01
  3   5  -6.057E+04   -5.300E+03    yes   2.873E-01  -9.578E-01   4.789E-01
  4   5  -6.791E+04   -5.941E+03    yes   2.873E-01  -9.578E-01   4.789E-01
  5   5  -8.565E+04   -7.493E+03    yes   2.873E-01  -9.578E-01   3.113E-01
  6   5  -6.748E+04   -5.903E+03    yes   2.874E-01  -9.578E-01   3.113E-01
  6   6  -5.444E+04   -4.762E+03    yes   2.873E-01  -9.578E-01   4.789E-01
  7   6  -6.165E+04   -5.394E+03    yes   2.873E-01  -9.578E-01   4.789E-01
  8   6  -5.419E+04   -4.741E+03    yes   2.874E-01  -9.578E-01   4.789E-01
  9   6  -7.100E+04   -6.212E+03    yes   2.873E-01  -9.578E-01   3.113E-01
  9   7  -5.014E+04   -4.387E+03    yes   2.873E-01  -9.578E-01   3.113E-01
 10   7  -4.527E+04   -3.961E+03    yes   2.873E-01  -9.578E-01   4.789E-01
 11   7   0.000E-01    0.000E-01    yes   0.000E-01   0.000E-01   0.000E-01
                    Side-B
                    ------
  i   j  normal str    shear str    slip?       unit normal              length
 11   9  -6.542E+04   -5.724E+03    yes  -2.873E-01   9.578E-01   2.390E-01
 10   9  -6.613E+04   -5.786E+03    yes  -2.873E-01   9.578E-01   3.113E-01
  9   9  -5.012E+04   -4.385E+03    yes  -2.873E-01   9.578E-01   3.113E-01
  9   8  -4.710E+04   -4.127E+03    yes  -2.873E-01   9.578E-01   4.789E-01
  8   8  -5.439E+04   -4.758E+03    yes  -2.873E-01   9.578E-01   4.789E-01
  7   8  -6.158E+04   -5.388E+03    yes  -2.874E-01   9.578E-01   4.789E-01
  6   8  -8.306E+04   -7.267E+03    yes  -2.873E-01   9.578E-01   3.113E-01
  6   7  -6.769E+04   -5.922E+03    yes  -2.873E-01   9.578E-01   3.113E-01
  5   7  -5.644E+04   -4.937E+03    yes  -2.874E-01   9.578E-01   4.789E-01
  4   7  -6.809E+04   -5.958E+03    yes  -2.873E-01   9.578E-01   4.789E-01
  3   7  -9.156E+04   -8.011E+03    yes  -2.873E-01   9.578E-01   3.113E-01
  3   6  -7.330E+04   -6.413E+03    yes  -2.873E-01   9.578E-01   3.113E-01
  2   6  -6.163E+04   -5.392E+03    yes  -2.873E-01   9.578E-01   4.789E-01
  1   6   0.000E-01    0.000E-01    yes   0.000E-01   0.000E-01   0.000E-01
```

FLAC

APPENDIX E

Tension Cutoff

The tension cutoff logic has been changed for the Mohr-Coulomb and ubiquitous joint models. In the previous versions, the tensile strength was assumed to be zero. In Version 2.10, the tensile strength can be defined by the user through the **PROP** command. The default is that no tension cutoff is assumed. Therefore, the user must specify a tension strength (positive or zero) in order for the tension cutoff logic to be activated.

The logic for the Mohr-Coulomb model checks for tensile stresses which exceed the cutoff strength. If this condition occurs, the stress is cut to zero in that zone at that timestep. The code then assumes that the material in that zone has failed in tension and can no longer sustain tensile stresses in the direction of the major principal stresses. Thus, for successive timesteps, no tensile stress will be allowed in that zone.

For the ubiquitous joint model, the stresses perpendicular and parallel to the joints are determined. If the stress perpendicular to the joint for the zone overcomes the tension cutoff, it is cut to zero and is unable to sustain tension in that direction after this time. If the average perpendicular stress component from the triangular subzones which make up a quadrilateral zone is tensile at any time, the joint cohesion is cut to zero for the zone, and remains zero thereafter.

F L A C

A P P E N D I X F

GROUNDWATER FLOW AND CONSOLIDATION IN FLAC

F1. INTRODUCTION

FLAC Version 2.2 models the flow of groundwater through a permeable solid, such as soil. The flow modeling may be done by itself (independently of the usual mechanical calculation of FLAC), or it may done in parallel with the mechanical modeling, so as to capture the effects of fluid/solid interaction. One type of fluid/solid interaction is "consolidation," in which the slow dissipation of pore pressure causes displacements to occur in the soil. This type of behavior involves two mechanical effects. First, changes in pore pressure cause changes in effective stress, which affect the response of the solid—for example, a reduction in effective stress may induce plastic yield. Second, the fluid in a zone reacts to mechanical volume changes by producing pore pressures, since the fluid is quite stiff.

At present, the code only handles fully-saturated flow, with phreatic surfaces represented crudely with a zero-pressure cut-off for pressures that try to become negative.

The user is strongly encouraged to become familiar with the operation of FLAC for simple mechanical problems before attempting to solve problems in which the flow and mechanical effects are both important.

After the section defining the input instructions for the groundwater option, a description of the solution scheme in FLAC is given with reference to two examples. These examples illustrate the use of most of the commands for the groundwater option.

F2. UNITS OF MEASUREMENT

Several different systems of measurements are in use for flow calculations. In addition, different terminology is used by different groups to refer to similar quantities. FLAC will accept any consistent set of units because the program makes no conversions internally. (FLAC does have a default fluid bulk modulus of 2e9, which is the modulus for water in Pascals, but this can be changed by the user.) This section is offered as a guide to converting between various sets of units and the corresponding quantities in FLAC.

The quantity most prone to cause confusion is the permeability. FLAC requires this to be in units of $[L^3T/M]$ (e.g., m^3s/kg). This rather curious unit is obtained because the program defines permeability as k/γ_w, where k is the hydraulic conductivity and γ_w is the unit weight of water ($g\rho_w$, where g is gravity and ρ_w is fluid density). k has units of velocity $[L/T]$, and g and ρ_w have units of acceleration $[L/T^2]$ and $[M/L^3]$, respectively. The overall units of permeability thus become

$$\frac{[L/T]}{[L/T^2]\ [M/L^3]}$$

These are <u>not</u> the permeability units normally used in soil mechanics. Some typical conversions between different units are given below.

1 darcy = 9.87×10^{-9} cm^2

1 cm/sec = 1.02×10^{-5} cm^2 (water at 20°C)

1 m/sec = 1.02×10^{-3} cm^2

Thus, if SI units are being used, the permeability required by FLAC is

$$k_{FLAC} = \frac{k}{g\rho_w} = \frac{(\text{permeability in } cm^2)/1.02 \times 10^{-3}}{9.87\ m^2/s \times 1000\ kg/m^3}$$

$$= \frac{\text{permeability in } cm^2}{10.07}$$

Thus, the permeability input to FLAC is

k_{FLAC} = (permeability in cm^2)/10.

When modeling a problem in fluid/solid interaction with FLAC, other physical constants must be specified. For an elastic soil, the <u>drained</u>

bulk and shear moduli, K and G, are needed. The porosity n of the soil is needed, as well as the apparent bulk modulus of the pore fluid, K_w. The maximum value that should be used for K_w is 2×10^9 Pa; normally, a lower value is used, to take account of the presence of trapped air in the pore space. The elastic constants are related to the confined compressibility M_c as follows:

$$1/M_c = K + (4/3)G$$

For zones below the water table, the density given for the soil is the saturated density—i.e.,

$$\rho = \rho_s + n\,\rho_w$$

where ρ_s = dry density of solid, and

ρ_w = density of fluid,

F3. INPUT COMMANDS

The commands necessary to model groundwater flow and consolidation are all implemented by means of added keywords on commands that are already in the standard FLAC code. Some duplication with the standard FLAC manual occurs here to make this appendix complete. This section simply defines the commands. The following sections demonstrate their use, and give a more detailed description of their function.

The commands used for groundwater flow are as follows.

APPLY keyword=value <range>

The available keywords are:

DISCHARGE
: A flux is specified (a volume rate per area (with units of cubic meters per square meter—i.e, m/s, for example)) over the range of GRIDPOINTS specified. This command is used to specify a constant inflow or outflow over a boundary of the grid.

WELL
: A volume rate of flow (e.g., cubic meters/ second) is specified for each ZONE in the range. This command can be used to represent either sources or, with a negative value, sinks. (Unit depth is assumed.)

CONFIG The keyword GW reserves memory for FLAC to run fluid flow problems. This command must be given before the grid is specified.

FIX **PP** <range>

The keyword PP causes the pore pressure to be fixed at the current value at gridpoints in the specified range. With this condition on a boundary, flow may occur, but the amount of flow is not specified.

FREE PP <range>

This is the reverse of the **FIX** command. If the pore pressure is free to change on a boundary gridpoint, and no **APPLY** command affects the gridpoint, no flow occure from that boundary

PLOT The following keywords can be used.

FLOW	plots flow vectors.
HEAD	plots contours of head (will only work if the **SET** command has been used to define gravity, and the water density is specified).
PP	pore pressure contours

PROP The following properties can be specified.

POROSITY	porosity
PERMEABILITY*	the isotropic permeability
K11,K12,K22*	components of the permeability tensor

*See Section F2 for units.

SET The following keywords apply to groundwater flow modeling.

DATUM=value <x1,x2 y1,y2>

This command can be used to set up the datum that FLAC uses for head calculations for the **PLOT HEAD** command. Different values can be specified for different parts of the grid, depending on the values of x1,x2,y1 and y2. If these are omitted, the entire grid is assumed. This command has no effect on the calculation, it only affects plotting and printing. (The default datum is the bottom of the grid.)

FLOW ON/OFF

This keyword turns the flow calculations on or off.

MECH ON/OFF

This keyword turns the mechanical calculations on or off.

SET (continued)

NGW=value

The number of groundwater flow steps done during each step is specified.

NMECH=value

The number of mechanical flow steps done during each step is specified.

WATER keyword=value

BULK	bulk modulus (default is 2e9)
DENSITY	water density
TABLE	The water table is input by the user with the **TABLE** command. This command tells FLAC which table to use.

F4. MODELING GROUNDWATER FLOW WITH FLAC

Many analyses of consolidation assume that water is incompressible in comparison with the deformability of the soil. FLAC, in contrast, assumes that fluid has a finite bulk modulus. For many soils, the solid bulk modulus is not very much lower than that of water. Pure water has a bulk modulus of 2×10^9 Pa. Normally, the pore space in a granular material is not completely filled with water—even a small percentage of trapped air can reduce the apparent bulk modulus of the water by an order of magnitude.

It is important to be clear about the time scale associated with consolidation and mechanical loading. The mechanical effects occur almost instantaneously in the real world—in the order of milliseconds. However, fluid flow is a long-term process: the dissipation associated with consolidation takes place in days, weeks, or even years. Real time is only accumulated when the flow option of FLAC is active.

As discussed in Section 3.0 of the FLAC manual, FLAC is an explicit code, which means that it takes "timesteps" to solve a problem. Thus, although mechanical effects take place almost instantaneously (e.g., pore pressure are generated when a load is applied), FLAC takes a finite (and sometimes large) number of steps to solve the problem. However, there is no true time period associated with these steps; they are merely an internal mechanism for the code to attain equilibrium. An alternative way to think of these mechanical steps is to imagine that each step represents a microsecond or less of

time, so that even if many steps are taken, almost no time elapses. To model this instantaneous development of pore pressure, the command **SET FLOW OFF** can be issued prior to the **SOLVE** command. This prevents flow and, hence, dissipation from occurring. (It is equivalent to setting the timestep to zero.) Once equilibrium is attained, examination of the pore pressures would show high concentrations under surface loads. In reality, these pore pressures now begin to dissipate. This can be modeled by turning the flow on with the command **SET FLOW ON** and again using the **SOLVE** or **STEP** command. The **SOLVE** command will model the real dissipation. Each timestep represents an increment of real time, as FLAC solves the transient fluid flow problem. The pore pressure concentrations will decrease, and settlement will occur under load.

In the previous paragraph, it was noted that, initially, many mechanical steps had to be taken, although time was not evolving. When fluid flow is occurring, mechanical effects are still observed (e.g., as fluid flows away, settlement occurs). FLAC may need to take several mechanical steps each time a fluid flow timestep occurs, just to bring the system back into mechanical equilibrium. (These mechanical steps are still just an internal device FLAC uses to maintain equilibrium.) Depending on how fast the system is changing, the number of mechanical steps necessary to maintain equilibrium can change. An indication of the number of mechanical steps required is the "out-of-balance force". As discussed in Appendix B of the FLAC manual, equilibrium is indicated by a low value, where "low" means "small relative to other

forces in the problem". If too few mechanical steps are being taken for each timestep, high out-of-balance forces will be observed, and the number of mechanical steps should be increased.

The **SET NMECH=value** command is used to specify the number of mechanical steps for each flow timestep. The default for NMECH is 1. Very late in a problem solution, the reverse may occur. Fluid dissipation may be so slow that a mechanical adjustment is only needed after several flow timesteps. In this case, NMECH should be set to 1, and the command **SET NGW=value** can be used, specifying the number of fluid steps before a mechanical adjustment is made. The default for NGW is 1. The following examples clarify the use of the **SET NMECH**, **SET NGW** and **FLOW** keywords. In all cases, the user is assumed to have issued the command **STEP 2**, and the sequence followed by FLAC is given.

1. **NMECH=1, NGW=1, FLOW=ON** (the default)

   Sequence: 1 mechanical step
   1 fluid flow timestep
   1 mechanical step
   1 fluid flow timestep

2. **NMECH=1, NGW=1, FLOW=OFF**

   Sequence: 2 mechanical steps

3. **NMECH=5, NGW=1, FLOW=ON**

   Sequence: 5 mechanical steps
   1 fluid flow timestep
   5 mechanical steps
   1 fluid flow timestep

4. **NMECH=1, NGW=5, FLOW=ON**

   Sequence: 1 mechanical step
   5 fluid flow timesteps
   1 mechanical step
   5 fluid flow timesteps

The following examples illustrate these commands.

F4.1 One-Dimensional Consolidation

The example of one dimensional consolidation is given in many textbooks on soil mechanics [e.g., Atkinson and Bransby (1978)]. A column of saturated soil of height H is impermeable at its base and free-draining at the top. A stress of σ_o is suddenly applied to the top surface. Instantaneously, the pore pressure in the column increases. If the water is incompressible, all of the applied load is carried by the fluid. As time goes by, the fluid flows out from the top of the column, and the pressure in the column decreases. Gradually, the load is transferred from the fluid to the solid. The theory of consolidation predicts the variation of pore pressure in time and space (1-d space).

If water is considered incompressible, the equation for excess pore pressure, u, as a function of depth z and time t is (Atkinson and Bransby, 1978):

$$c_v \frac{\partial^2 u}{\partial z^2} = \frac{\partial u}{\partial t} \qquad \text{(F-1)}$$

where $c_v = \frac{k}{M_c \gamma_w}$, with the dimensions L^2/T, the coefficient of consolidation,

k is the hydraulic conductivity in L/T units,

M_c is the confined, drained compressibility of the soil, and

γ_w is the unit weight of water.

If water is considered to be compressible, with a bulk modulus of K_w, Eq. (F-1) still applies, but c_v is now

$$c_v = \frac{k}{M_c + (n/K_w)} \qquad \text{(F-2)}$$

where n is the porosity, and

$k_{FLAC} = k/\gamma_w$, which is the permeability required by FLAC.

Obviously, as K_w tends to infinity, Eq. (F-2) reduces to Eq. (F-1). Equation (F-2) is a more general expression of c_v, and is the one that applies to one-dimensional consolidation modeled with FLAC. There are other consequences that follow from the assumption of compressible water: for example, the instantaneous excess pore pressure u_I is less than the applied stress, σ_o, in the odometer test—the ratio is given by

$$\frac{u_I}{\sigma_o} = \frac{K_w}{K_w + (n/M_c)} \qquad \text{(F-3)}$$

Recall from Section F2 that

$$1/M_c = K + (4/3)G \qquad \text{(F-4)}$$

If c_v and u_I/σ_o are known from an odometer test, the required constants for FLAC can be found by using Eqs. (F-2, F-3 and F-4).

The following sequence of input commands to FLAC causes it to model the one dimensional consolidation of a 20 m column.

```
config gw
* 1-D consolidation test
grid 1 20
model elastic
fix x           * (just allow vertical motion)
fix x y j=1     * (fix the base mechanically)
prop d 2000 bulk 5e8 shear 2e8
prop perm=1e-10
water bulk=2e9
* all zone pore-pressures are equal
  ini pp=1e5
* complete drainage at top surface
* (this will reduce top-zone pressure)
  ini pp=0 j=21
  fix pp j=21
* set total stresses to equal applied load
  apply pres=1e5 j=21
  ini syy=-1e5
his pp i=1,j=10
his ydis i=1 j=21
set nmech=25
step 500
set nmech 15
step 300
set nmech 10
step 200
pr gpp
set nmech=5
step 1000
pr gpp
step 1000
pr gpp
set nmech 2
step 1000
pr gpp
step 1000
pr gpp
```

The column of 20 zones is 20 meters in height. There is no gravity, since we are only interested in the excess pore pressure effect. The system is in mechanical equilibrium initially, with the applied surface stress equal to the internal total stress of 10^5 Pa. Since the pore pressure is also set to the same value, the initial effective stress is zero. The system does not remain in equilibrium because the pore pressure at the top gridpoint is set to zero—water starts flowing upwards and reduces the total stress, which causes the soil to start compacting. FLAC chooses the fluid timestep as a function of the properties and geometry. In this case, because of the choice of properties, one step corresponds to one second of time. The complete distribution of pore pressure is printed out every 1000 seconds. The table below compares FLAC's results with the exact solution of Eq. (F-1), using c_v from Eq. (F-2). The pressures are expressed as fractions of the initial pressure, and are given at two locations: the base, and halfway up the column.

TIME	P AT BASE		P at 10 meters	
(sec)	Exact	FLAC	Exact	FLAC
1000	0.885	0.885	0.653	0.651
2000	0.642	0.644	0.455	0.457
3000	0.457	0.457	0.323	0.323
4000	0.324	0.324	0.229	0.229
5000	0.230	0.230	0.163	0.163

The various commands used in the example are defined in Section F2, above. Some discussion of those used in this example follows.

If a FLAC run is to incorporate interaction between groundwater and a deformable porous solid, then a **CONFIG GW** command must be given prior to the **GRID** command, as in this example. This allocates the memory necessary to hold the extra variables. By default, full interaction will be assumed when this **CONFIG** command is given. However, the run may subsequently be done with or without groundwater interaction, as discussed below.

The **WATER** command is used to specify the apparent bulk modulus and mass density of water: K_w and ρ_w, respectively. The keywords are BULK and DENSITY. The density is only used if gravity is specified. Under hydrostatic conditions, the pore pressure gradient will be $\rho_w g$, where g is the gravitational acceleration. Note that the density of the solid (specified with the **PROP** command) is the saturated density.

The **BULK** parameter of the **WATER** command sets the bulk modulus of the fluid. It should not be set unrealistically high if "undeformable" fluid is required; rather, it should be kept at the correct value, and the bulk modulus of the solid reduced. The bulk modulus of water in S.I. units is 2×10^9 Pa. The bulk modulus may be set to zero if the **FLOW OFF** condition is in effect (see discussion below) but, otherwise, a non-zero value is required to compute a finite fluid timestep.

In FLAC's input and output, pore pressure is expressed as a true pressure (e.g., in Pascals), rather than a head, which is more commonly used in soil mechanics. One advantage of this is

that disconnected groundwater regions can exist without ambiguity in the specification of a reference point for heads. Pore pressure is an exception to the usual FLAC rule in which scalars are associated with zones. For the groundwater flow calculation, pressure is assumed to be located at gridpoints—zone pressures are then derived from these "master" values by averaging; hence, the gridpoint values are more accurate. They may be printed by **PRINT GPP**. Zone values are printed if **PRINT PP** is given. On plots of pressure, however, the gridpoint values are always used.

In general, permeability may be given as an anisotropic tensor, by means of its components k_{11}, k_{12} and k_{22}, referred to the coordinate axes (Note that $k_{12}=k_{21}$.). Mohr's circle can be used to convert between these components and the principal values and their angles. The permeability used by FLAC is defined as follows:

$$v_i = (k_{ij})_{FLAC} \frac{\partial p}{\partial x_j} \qquad \text{(F-5)}$$

where v_i is the specific discharge vector (flow rate per unit area).

To convert to FLAC's units from the velocity units of k, used in Eq. (F-1), divide by the unit weight of water, γ_w:

$$k_{ij_{FLAC}} = \frac{k_{ij}}{\gamma_w} = \frac{k_{ij}}{g\rho_w} \qquad \text{(F-6)}$$

FLAC never does this calculation automatically, even if the density is given on the **WATER** command.

If the keyword **PERM** is used on the **PROP** command, both components $(k_{11})_{FLAC}$ and $(k_{22})_{FLAC}$ are set to the given value, and $(k_{12})_{FLAC}$ is set to zero (i.e., an isotropic permeability).

Keywords K11, K22, K12 are used to specify the components of an anisotropic permeability tensor. Note that a high permeability will result in a small fluid time-step. Instead of using high-permeability regions, it is often possible to specify, instead, a fixed pressure around the boundary of the region using the **FIX PP** and **INI PP** commands. This allows free drainage.

When the **INI** command is used to specify initial and boundary pore pressures, the given values are installed in the gridpoints covering the prescribed range. Fixed-pressure conditions are set by **FIX**ing the pressures at the required gridpoints; fluid may flow into and out of the grid at these points. Boundaries that do not

have the **FIX** condition set are impermeable. Note that the **FREE** command may be used to revoke the **FIX** condition.

Care must be taken when prescribing initial pressure distributions and boundary pressures: if gravity exists, then pressure gradients must be given, via the VAR parameter with the **INI** command.

The commands **SET NMECH** in the input file specify how many mechanical steps are taken for each fluid flow timestep. At the start of the run, there is a large mechanical imbalance, so NMECH is set to 25 in order for the grid to settle down mechanically between timesteps. NMECH is then reduced as the disturbing effects become less, due to the reduced changes in pore pressures. In this example, larger values of NMECH are used than are probably necessary in later stages of the problem; if the values are too small, inertial effects (oscillations or lagging response) will be observed in the pressure-time histories. Some experimentation is usually necessary to find the most efficient solution. If this problem is run, it will be observed that, near the beginning of the problem, the out-of-balance force is about 400 Pa or more; by the end, however, it has reduced to less than 50 Pa. If the same problem is run with NMECH=1 initially, the unbalanced forces start out at several thousand Pascals. In this example, a "representative" force would be 10^4 to 10^5 (typical zone dimension (1 m) multiplied by typical stress, 10^5 Pa), so a "reasonable" out-of-balance force would be about 1% of this (i.e., 100 to 1000).

Several other options not illustrated in this example are available for fluid flow calculations. These are:

SET FLOW OFF	eliminates calculation of dissipation. Pore pressures change due to mechanical loading, but no flow occurs.
SET MECH OFF	eliminates the mechanical calculation, used to solve a flow problem only.
SET NGW = nstep	runs nstep groundwater steps for one mechanical step. This option can be used when the solution is almost at a steady state. It is equivalent to being able to reduce NMECH below 1.
WATER BULK=0 & SET FLOW OFF	eliminates all fluid effects-generation and dissipation.

F4.2 A Two-Dimensional Example

This example illustrates the use of FLAC to compute the flow field under a sheet pile, which is represented as a vertical line of zero-permeability zones. Upstream, we have a 10 m head; downstream we have a 5 m head. No mechanical effects are modeled, and gravity is ignored. Pressures on the ground surface are fixed (as this boundary is in direct contact with the water), and other boundaries are impermeable.

```
*  flow underneath a sheet pile; no mechanical
*  effects
config gw
g 21,10
m e
prop d 2000 s 1e9 b 2e9 perm=1e-14
* effect of a sheet-pile ...
prop perm=0 i=11 j=6,10
water bulk=2e9 dens=1000
fix x y j=1
fix x i=1
fix x i=22
fix pp j=11
* upstream side ------------
ini pp=1e5   i=1,11  j=11
* downstream side ----------
ini pp=0.5e5 i=12,22 j=11
* switch off mechanics ------
set mech=off
* solve ...
his pp i 5 j 5
step 1000
plot pen b pp
ret
```

Plotter output for this data file is shown as Fig. F-1.

Fig. F-1 Pore Pressure Contours Around a Sheet Pile (no gravity or mechanical effects)

With gravity and mechanical interaction, things become more complicated. When the pore pressure at a boundary is specified, this represents only a fluid condition; no mechanical loading is implied. If mechanical loading from a standing head of water is desired, the appropriate **APPLY** command must also be given. The previous example is repeated, but with gravity and mechanical loading added. The following data file produces the stress and pore-pressure distributions shown in Figure F-2.

```
*  flow underneath a sheet pile with mechanical
*  interaction
config gw
g 21,10
m e
prop d 2500 s 1e9 b 2e9 perm=1e-14
* note: density includes water
* effect of a sheet-pile ...
prop perm=0 i=11 j=6,10
water  dens=1000
fix x y j=1
fix x i=1
fix x i=22
fix pp j=11
* upstream side ------------
ini pp=1e5     i=1,11  j=11
apply pres=1e5 i=1,11 j=11
* downstream side ----------
ini pp=0.5e5     i=12,22 j=11
apply pres=0.5e5 i=12,22 j=11
set grav=10
his pp i 5 j 5
his yd i 5 j=11
step 1000
save sheet1.sav
ret
```

Fig. F-2 Pore Pressure and Stress Distribution Flow Under a Sheet Pile (with gravity and mechanical effects)

Once equilibrium of flow and stress has been obtained, a load can be placed on the ground surface representing a structure, for example. The response to this is in two parts. First, an "instantaneous" adjustment in stresses and pore pressures occurs: for this stage, the flow calculation is switched off. Second, slow dissipation of pore pressure takes place: in this phase, the time scale is "real." The first stage is accomplished by the following data file.

```
restore sheet1.sav
hist reset          * forget the old histories
hist pp   i=5 j=5
hist ydis i=11 j=11
* apply extra mechanical loading centered on
* sheet-pile
apply pres=5e5 i=9,14 j=11
* get to 'instantaneous' mechanical equilibrium
* first:
set flow=off
step 750
save sheet2.sav
ret
```

The resulting redistribution of pore pressure and stress is observed in Figure F-3.

Fig. F-3 Redistribution of Stress and Pore Pressure When Load Is Applied

Now, the flow calculation is turned on, and "real" time is allowed to elapse. After executing the following data file, the history of pore pressure dissipation shown in Fig. F-4 is produced. The field of consolidation displacements is illustrated in Fig. F-5.

```
restore sheet2.sav
hist reset
ini xd=0 yd=0
* future displacements will be w.r.t. this state
hist yd i 11 j 11
hist pp i 5 j 5
set flow=on     * restore flow capability
* allow pp to dissipate
step 1000
save sheet3.sav
ret
```

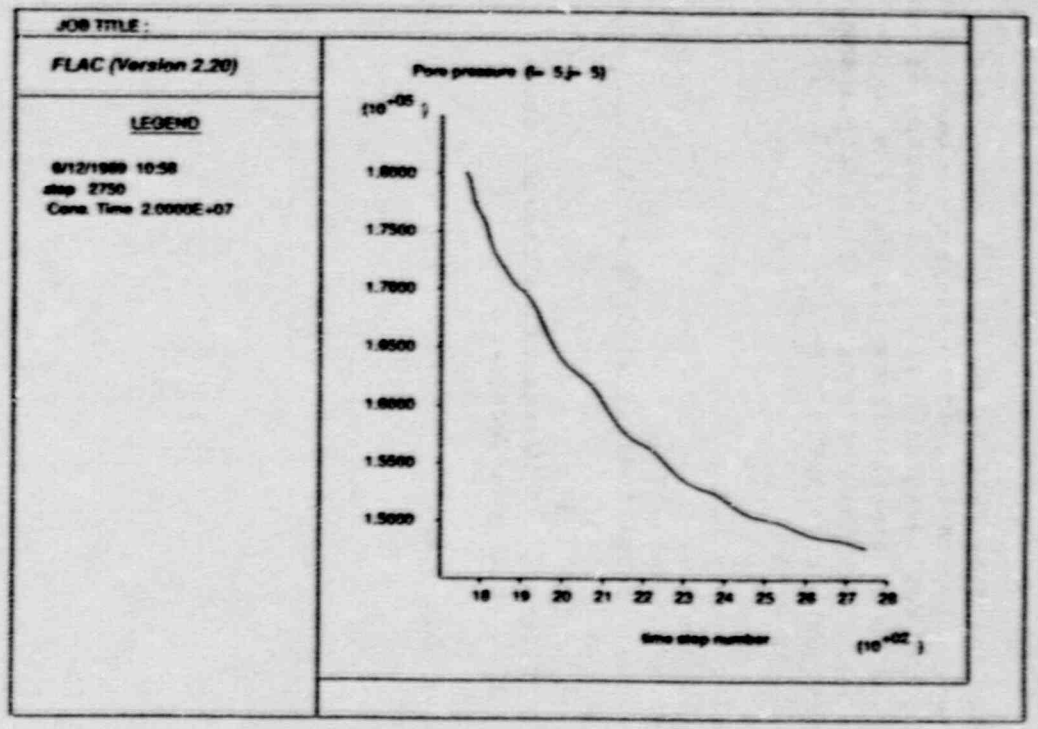


Fig. F-4 Pore Pressure Dissipation Under Applied Load As a Function of Time

Fig. F-5 Consolidation Displacements After Drainage Has Occurred

Figure F-4 shows some oscillation in the pore pressure history, indicating that, perhaps, an insufficient number of mechanical steps were run (NMECH=1, the default value, in this problem).

A typical out-of-balance force in this problem is about 5e5 Pa, so a small value would be about 5e3. When the problem is run, the unbalanced force starts out at 10^4 and drops to about 10^3 over 250 steps. Finally, after 1000 steps, it is about 10^2. This indicates that, initially, NMECH probably should have been higher.

If the **STEP 1000** command is replaced by the sequence

```
SET NMECH=10
STEP 500
SET NMECH=5
STEP 300
SET NMECH=2
STEP 200
```

the pore pressure evolution shown in Fig. F-6 is obtained. The curve is much smoother, but the overall shape probably is sufficiently similar to Fig. F-4 to accept the original results. In this problem, the unbalanced force drops to less than 10^3 in about 50 steps, and is less than 10^2 after 250 steps.

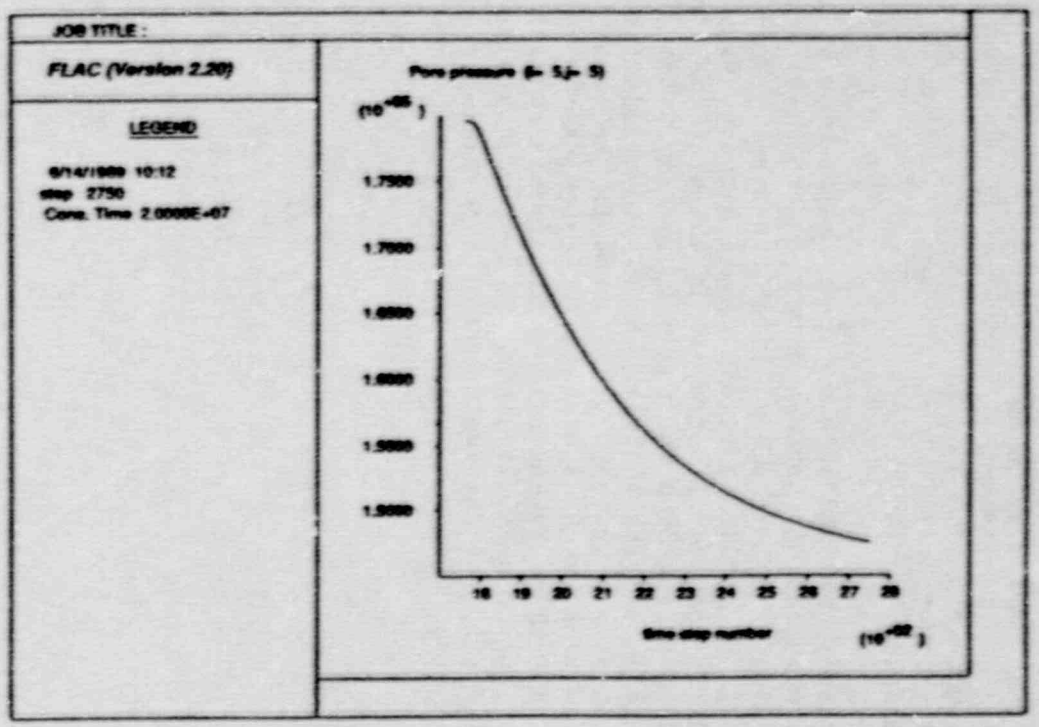


Fig. F-6 Pore Pressure History with More Mechanical Steps

If the command sequence

```
SET NGW=10
HIS NSTEP=1
STEP=100
```

replaces the **STEP** 1000 command, the problem will run much faster, because only one mechanical step is done after every ten fluid flow steps. However, the resulting pore pressure history (Fig. F-7) is noticeably different in shape to that in FIg. F-6, although the curve is smooth. The out-of-balance force in this case is about 5e4 after 25 steps (equivalent to 250 steps in the other cases), and remains above 2e4 throughout the 1000 steps. These out-of-balance forces are excessive for this problem, resulting in an inaccurate representation of the evolution of pore pressure with time.

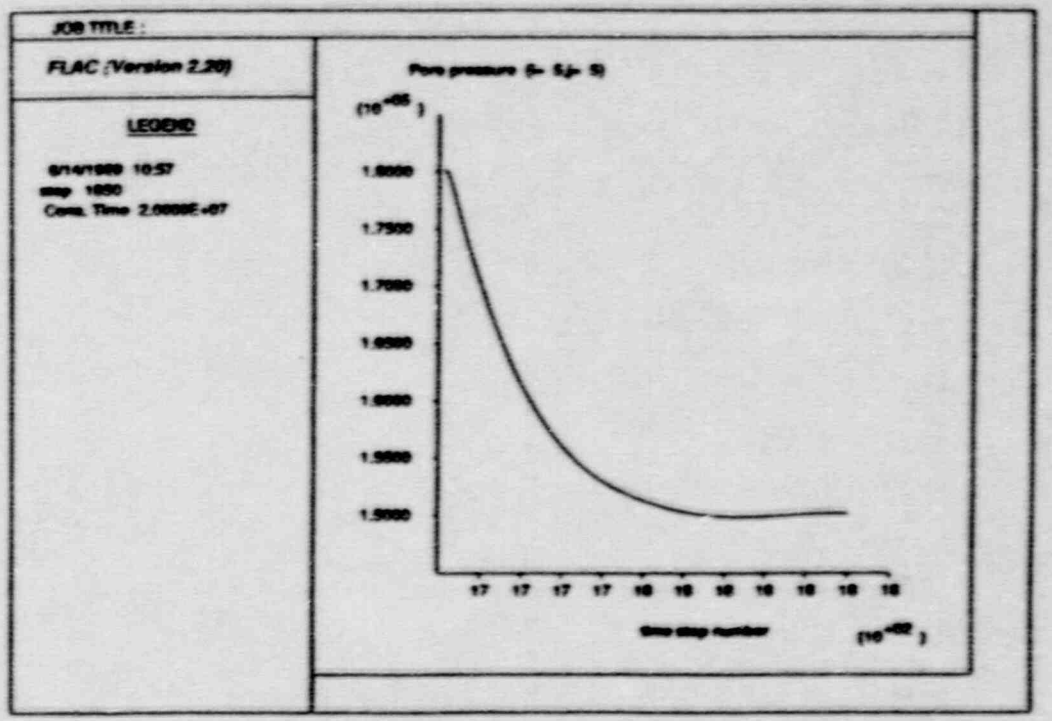


Fig. F-7 Pore Pressure History with Too Few Mechanical Steps

F5. REFERENCES

Atkinson, J. H. and P. L. Bransby. The Mechanics of Soils — An Introduction to Critical State Soil Mechanics. New York: McGraw-Hill, 1978.

F L A C

A P P E N D I X G

GRAPHICS ON DOT-MATRIX PRINTERS AND THE HP-LASERJET

Many users have requested the ability to send FLAC graphics to devices other than HP plotters and PostScript laser printers. Version 2.2 offers the ability to send graphics to dot-matrix printers, such as Epson, Panasonic, IBM, as well as the HP Laserjet. The printer drivers we use are produced by a third party, so Itasca cannot be responsible for failure to work on specific printers. We have only been able to test this feature on a limited number of printers. So far we have successfully used an NEC 890 (emulating an HP Laserjet), a Panasonic 1080, Epson MX-80, LQ-800 and LX-80 printers, and the HP Paintjet printer. (All printers produce black and white output only.)

Also, the plots can only be produced by a post-processing operation. A machine with an 8087, 80287 or 80387 coprocessor is required for the postprocessing. (Users with a Weitek coprocessor will have to do the plotting from a machine with one of these chips unless they also have an 80x87 chip.)

The first step is to install the files you will need on your hard disk. First, consult the attached list of printers and select the appropriate xxxxx.PRT file for your printer. Then, go to the directory from which you will be making plots, copy the program INSTALL.BAT to your hard

disk and, with a copy of the FLAC disk in the A drive, type INSTALL xxxx, where xxxx is replaced by the name you selected for your printer. This program will do the following.

1. Copy the following files to your hard disk.

   "NORMAL.FNT",
   "ITLBLD.FNT",
   "DPRINT.EXE",
   "INTERP.EXE"
   "POP.BAT"

2. Copy the appropriate driver for your printer under the name DPRINT.PRT.

At this point, your computer is set up to plot to the printer you selected. If you ever wish to change printer, repeat this process for the new printer. If you wish to use more than one printer on a regular basis, rename the file "DPRINT.PRT" to something else, and repeat the INSTALL process for the other printer. You may have as many .PRT files present as you wish, but the plot program will always use the one named "DPRINT.PRT".

To generate the printer plot, follow these steps.

In FLAC

(1) Use the command **SET PLOT=PRINTER** to tell the program you want output for a printer rather than the HP plotters or PostScript printers.

(2) Use the command **SET OUT=fname** to specify the file to which you want the output sent. If this file already exists, it will be overwritten.

(3) Generate the plot you desire, using **PLOT PEN** .....

(4) If you want more plots, specify a new filename for each one using the **SET OUT=fname**, as in (2), above. The previously selected file will then be closed and a new one opened.

(5) When you are finished using FLAC, get out by typing **STOP** or **QUIT**.

Out of FLAC

Use the Printer Output Program (POP) to generate your figure as follows. In the directory where you installed your output programs, type **POP fname**, where fname is the file you generated in FLAC. The program will then proceed to process this file to produce the plot you requested.

WARNING: As part of this process, the files "PLOT.TMP" and "PLOT.PLT" are generated and erased. If you have any other file with one of these names, rename it or it will be destroyed!

Producing plots with this process is slow, especially if more than one plot is being made. If you wish to create a batch file to produce several plots without your intervention, create a file similar to "POP.BAT", but with your filenames as the input to INTERP.

NOTE: The POP program runs the DOS program GRAPHICS. This program is a part of DOS, and must be accessible from the directory where you are producing plots. For some printers, this program is not necessary, so you may wish to try removing it from POP and see if your printer works.

Modifying Printer Output

The printer drivers we supply are supposed to be device-independent, so that everything appears the same size regardless of printer type or resolution. In reality, we have found that text size, in particular, varies. The file that FLAC produces is a standard ASCII file and can be edited, just like a FLAC data file. This section explains the format of this file, so that you can adjust text size, if you wish. Be careful — It is possible to change the plot completely by changing one number incorrectly.

Each command consists of a pair of lines, except that text input takes three lines. The first is a command number, and the second is a list of parameters for the command. The command numbers are as follows.

1 - initialize graphics
2 - text
3 - filling (disabled for dot-matrix printers)
4 - draw a line
5 - specify character size
6 - specify font for text

One parameter you may want to change is the text size. To do this, find a line with a single 5 in it, followed by a line with a single real number. This number is the approximate text size in inches. The particular size command you are interested in will be the one immediately preceding the text whose size you want to change. It can be adjusted until you are satisfied.

For example, the lines

```
5
.7500E-01
```

attempt to set up text with a height of 0.75 inches. Future text in the file will be of this size until another pair of lines, such as

```
5
.7000E-01
```

occurs.

We have also found that some 24-pin dot matrix printers, such as the Epson LQ-800, produce plots with the incorrect aspect ratio (i.e., square zones appear as rectangles). This can be corrected by modifying the graphics initialization (the first command in your file) as follows: replace the third parameter (.8412e+01) by a smaller number until squares appear square. To calculate the new parameters, use the equation

$$p^{new} = 1.0 + 7.412 \times (y_{paper}/x_{paper})$$

where p^{new} is the new third parameter, and x_{paper}, y_{paper} are the dimensions of a square zone on your printer.

We hope to improve the printer output program in the future, so we invite you to send examples of output from your printer (please let us know which) and any comments on the program.

Table G-1

PRINTER DRIVERS AVAILABLE WITH FLAC

Manufacturers Models	Printer Driver	Printer Driver	Resolution Vertical	Resolution Horizontal	Color Range
Alphacom (E)	Aero	EPSOBL.PRT	60	72	0-1
		EPSOBH.PRT	120	72	0-1
		EPSOHH.PRT	120	216	0-1
AMT	Office Printer	AMTVL.PRT	60	60	0-1
		AMTL.PRT	120	60	0-1
		AMTH.PRT	120	120	0-1
		AMTH.PRT	240	120	0-1
		AMTVH.PRT	240	240	0-1
		AMTV/H.PRT	480	240	0-1
		AMTCVL.PRT	60	60	0-15
		AMTCL.PRT	120	60	0-15
		AMTCH.PRT	120	120	0-15
		AMTCH.PRT	240	120	0-15
		AMYCVH.PRT	240	240	0-15
		AMICVVH.PRT	480	240	0-15
Anadex	WP Series	ANADEIL.PRT	72	72	0-1
		ANADEIH.PRT	144	144	0-1
		ANADEICL.PRT	72	72	0-15
		ANADEICH.PRT	144	144	0-15
	DP Series	DATASTHL.PRT	72	72	0-1
		DATASTHH.PRT	144	144	0-1
Antex Data Systems (E)	ADS 2000	EPSOBL.PRT	60	72	0-1
		EPSOBH.PRT	120	72	0-1
		EPSOHH.PRT	120	216	0-1
AT&T	475	CITOHL.PRT	80	72	0-1
		CITOHH.PRT	96	72	0-1
		CITOHH.PRT	136	72	0-1
Brother	2024L, 3x density	LQ1300.PRT	180	180	0-1
	Twinriter 5 HP mode	TWIBL.PRT	60	72	0-1
		TWIHH.PRT	120	72	0-1
		TWIHH.PRT	120	216	0-1

Table G-1 — 2

Manufacturers Model	Printer Driver	Resolution Vertical	Resolution Horizontal	Color Range
CAL-ABCO (E)	Legend 1385	EPSOBL.PRT	60	72	0-1
	CP-VII	EPSOBH.PRT	120	72	0-1
		EPSOBH.PRT	120	216	0-1
Centronics	All	CENTDOD.PRT	60	60	0-1
Citizen (E)	MSP-10	EPSOBL.PRT	66	72	0-1
	MBP-25	EPSOBH.PRT	120	72	0-1
		EPSOBH.PRT	120	216	0-1
C.Itoh	8510, 8600	CITOBL.PRT	80	72	0-1
	Prowriter I & II	CITOBH.PRT	96	72	0-1
		CITOBH.PRT	136	72	0-1
		CITOBVH.PRT	160	72	0-1
Canon	PJ-1080A	CANOG.PRT	84	84	0-15
	PW-1156A	EPSOBL.PRT	60	72	0-1
		EPSOBH.PRT	120	72	0-1
		EPSOBH.PRT	120	216	0-1
Dataproducts	8030, 8070	DAT8030.PRT	160	84	0-1
	8052C	IBMCLBL.PRT	60	72	0-1
		IBMCLBH.PRT	120	72	0-1
		IBMCLBCL.PRT	60	72	0-15
		IBMCLBCH.PRT	120	72	0-15

Table G-1 — 3

Manufacturers Models	Printer Driver		Resolution Vertical	Horizontal	Color Range
Datasouth	All	DATASTHL.PRT	72	72	0-1
		DATASTHH.PRT	144	144	0-1
DEC	DECwriter IV LA100, LA50 LN03	DECL.PRT	144	72	0-1
		DECH.PRT	180	72	0-1
Diablo (E)	P series,34LQ	EPSONL.PRT	60	72	0-1
		EPSONH.PRT	120	72	0-1
		EPSONH.PRT	120	216	0-1
	332	DIABS32.PRT	70	70	0-1
Dynax-Fortis	DX45	TWINL.PRT	60	72	0-1
		TWINH.PRT	120	72	0-1
		TWINH.PRT	120	216	0-1
		TWINVH.PRT	240	216	0-1
Epson	FX series	EPSONL.PRT	60	72	0-1
		EPSONH.PRT	120	72	0-1
		EPSONH.PRT	120	216	0-1
		EPSONVH.PRT	240	216	0-1
	JX series	JX80L.PRT	60	72	0-15
		JX80H.PRT	120	72	0-15
		JX80H.PRT	120	216	0-15
		JX80VH.PRT	240	216	0-15
	RX series	STARL.PRT	60	72	0-1
		STARH.PRT	120	72	0-1
	LQ 1500	LQ1500.PRT	180	180	0-1
	LQ 2500	LQ1500.PRT	180	180	0-15
		LQ1500C.PRT	180	180	0-15

Table G-1 — 4

Manufacturers Models	Printer Driver	Resolution Vertical	Resolution Horizontal	Color Range
Facit	4528	FACI4528.PRT	60	60	0-1
	4542, 4544	FACI4542.PRT	70	70	0-1
Fujitsu	DL24B	FUJITSUL.PRT	60	60	0-1
		FUJITSUH.PRT	90	90	0-1
		FUJITSUW.PRT	180	180	0-1
		FUJITSCH.PRT	180	180	0-15
	DPL24C	LQ1500.PRT	180	180	0-1
		LQ1500C.PRT	180	180	0-15
JBL	750 series	JBL750L.PRT	60	60	0-1
		JBL750H.PRT	90	90	0-1
		JBL750H.PRT	180	180	0-1
		JBL750LC.PRT	60	60	0-15
		JBL750HC.PRT	90	90	0-15
		JBL750HC.PRT	180	180	0-15
CERICOR	3180-3404 Series	CERICOR.PRT	72	72	0-1
Corilla	All	CORILLA.PRT	60	63	0-1
Norsea (E)	Printer I	EPSOHL.PRT	60	72	0-1
		EPSOHH.PRT	120	72	0-1
		EPSOHH.PRT	120	216	0-1
Hewlett-Packard	LaserJet	HPLSH75.PRT	75	75	0-1
	LaserJet Plus	HPLSH100.PRT	100	100	0-1
	Series II	HPLSH150.PRT	150	150	0-1
		HPLSH300.PRT	300	300	0-1

Table G-1 — 5

Manufacturers	Models	Printer Driver	Resolution Vertical	Resolution Horizontal	Color Range
IBM	Color Printer	IBMCLRL.PRT	60	72	0-1
		IBMCLRH.PRT	120	72	0-1
		IBMCLRCL.PRT	60	72	0-15
		IBMCLRCH.PRT	120	72	0-15
	Graphics Printer	IBMGRPHL.PRT	60	72	0-1
	Proprinter	IBMGRPHH.PRT	120	72	0-1
		IBMGRPHH.PRT	120	216	0-1
		IBMGRPHV.PRT	240	216	0-1
	Color Jetprinter	IBMIJB&W.PRT	84	63	0-1
	3852	IBMIJCLR.PRT	84	63	0-15
IDS	560, 480, P132	IDSB&W.PRT	84	84	0-1
	P80	IDSCOLOR.PRT	84	84	0-15
	440	IDS440.PRT	64	64	
Integrex	Colour Jet 132	INTEGREX.PRT	60	60	0-1
Intoq	LP-5100A	DATASTHL.PRT	72	72	0-1
		DATASTHH.PRT	144	144	0-1
Malibu	165	MALIBU.PRT	60	60	0-1
Mannesmann Tally	160	MANN160L.PRT	50	64	0-1
		MANN160H.PRT	100	64	0-1
		MANN160H.PRT	133	64	0-1
	420, 440	MANN420.PRT	60	60	0-1
	Sprint 80	MTS80L.PRT	80	72	0-1
		MTS80H.PRT	160	72	0-1
		MTS80H.PRT	160	216	0-1
NEC	All	NPIL.PRT	60	72	0-1
		NPIH.PRT	120	72	0-1

Table G-1 — 6

Manufacturers Models	Printer Driver		Resolution Vertical	Resolution Horizontal	Color Range
		NPIN.PRT	120	144	0-1
NEC	P2, P3	NECP2P3L.PRT	60	60	0-1
		NECP2P3H.PRT	120	60	0-1
		NECP2P3H.PRT	120	120	0-1
		NECP2VH.PRT	240	240	0-1
	CP2, CP3	NECP2CL.PRT	60	60	0-1
		NECP2CH.PRT	120	60	0-1
		NECP2CH.PRT	120	120	0-1
		NECP2CVH.PRT	240	240	0-1
	8023	NEC8023.PRT	72	72	0-1
	8027A	NEI8027A.PRT	80	72	0-1
	P5, P6, P7	LQ1500.PRT	180	180	0-1
		NECP5VH.PRT	360	360	0-1
		LQ1500C.PRT	180	180	0-15
		NECP5CVH.PRT	360	360	0-15
North Atlantic Quantex	All	NRTHATQL.PRT	72	72	0-1
		NRTHATQH.PRT	120	72	0-1
		NRTHATQH.PRT	144	72	0-1
Okidata	ML-92, ML-93 ML-82 w/out P&P ML-84 w/out P&P ML-192, ML-193	OKIDATA.PRT	72	72	0-1
	Above, w/P&P	IBMGRPHL.PRT	60	72	0-1
		IBMGRPHH.PRT	120	72	0-1
		IBMGRPHH.PRT	120	216	0-1
	2410, 2350	OKI2410.PRT	72	72	0-1
Olympia (E)	NP	EPSONL.PRT	60	72	0-1
		EPSONH.PRT	120	72	0-1

Table G-1 — 7

Manufacturers	Models	Printer Driver	Resolution Vertical	Horisontal	Color Range
		EPSONH.PRT	120	216	0-1
Panasonic (E)	All	PANASL.PRT	60	72	0-1
		PANASM.PRT	120	72	0-1
		PANASH.PRT	120	216	0-1
		PANASVH.PRT	240	216	0-1
PMC	DRP-85	NEC8023.PRT	72	72	0-1
Quadram	Quadjet in color	QUADJETC.PRT	80	80	0-15
	Quadjet in b&w	QUADJET.PRT	80	80	0-1
Radio Shack	LP-VII	SEIKOSHA.PRT	60	63	0-1
	DMP-500,DMP-420	RADIOSHK.PRT	60	72	0-1
	DMP-430	RS430L.PRT	120	144	0-1
		RS430M.PRT	140	144	0-1
		RS430H.PRT	194	144	0-1
	DMP-2100	RS2100.PRT	100	100	0-1
	CGP-220 in color	RSCGP22C.PRT	80	80	0-15
	CGP-220 in b&w	RSCGP220.PRT	80	80	0-1

Table G-1 — 8

Manufacturers Models	Printer	Driver	Resolution Vertical	Horisontal	Color Range
Riteman (E)	All	EPSONL.PRT	60	72	0-1
		EPSONN.PRT	120	72	0-1
		EPSONN.PRT	120	216	0-1
Seikosha	GP-100A	SEIKOSHA.PRT	60	65	0-1
Smith-Corona (E)	D-200, D-300	EPSONL.PRT	60	72	0-1
		EPSONN.PRT	120	72	0-1
		EPSONN.PRT	120	216	0-1
Star Micronics	Delta, Radix	STARL.PRT	60	72	0-1
	Gemini,SG-15	STARN.PRT	120	72	0-1
	SD-15, SR-15	STARN.PRT	120	144	0-1
		STARVH.PRT	240	144	0-1
	SB-10 24 pin mode	STARSB10.PRT	180	240	0-1
Texas Instruments (E)	TI 850	EPSONL.PRT	60	72	0-1
	TI 855	EPSONN.PRT	120	72	0-1
		EPSONN.PRT	120	216	0-1
Toshiba	All except 1340 and 1350	TOSHIBA.PRT	180	180	0-1
		TOSHIBAN.PRT	360	180	0-1
	1350	TOSH1350.PRT	180	180	0-1
	COLOR	TOSHIBAC.PRT	180	180	0-15
		TOSHIBCH.PRT	360	180	0-15

F L A C

A P P E N D I X H

THERMAL-MECHANICAL OPTION

H1. INTRODUCTION

This option of FLAC simulates the transient flow of heat in materials and the subsequent development of thermally-induced stresses.

This option has the following specific features.

1. There are three material models for the thermal behavior of the material — isotropic, anisotropic conduction, and a temperature-dependent conductivity model.

2. As in the standard version of FLAC, different zones may have different models and properties.

3. Any of the mechanical models may be used with any of the thermal models.

4. Several different thermal boundary conditions may be prescribed.

5. Heat sources may be inserted into the material either as line sources or as volume sources. These sources may be made to decay exponentially with time.

6. Both explicit and implicit algorithms are available.

H2. THEORETICAL BACKGROUND

H2.1 Basic Equation

The basic equation of conductive heat transfer is Fourier's law, which can be written in one dimension as

$$Q_x = - k_x \frac{\partial T}{\partial x} \qquad \text{(H-1)}$$

where Q_x = flux in the x-direction (W/m^2), and

k_x = thermal conductivity in the x-direction ($W/m°C$).

A similar equation can be written for Q_y. Also, for any mass, the change in temperature can be written as

$$\frac{\partial T}{\partial t} = \frac{Q_{net}}{C_p M} \qquad \text{(H-2)}$$

where Q_{net} = net heat flow into mass (W),

C_p = specific heat (J/kg°C), and

M = mass (kg).

These two equations are the basis of the thermal version of FLAC.

Equation (H-2) can be written as

$$\frac{\partial T}{\partial t} = \frac{1}{C_p \rho} \left[\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} \right]$$

Combining this with Eq. (H-1),

$$\frac{\partial T}{\partial t} = \frac{1}{C_p \rho} \frac{\partial}{\partial x} \left[k_x \frac{\partial T}{\partial x} \right] + \frac{\partial}{\partial y} \left[k_y \frac{\partial T}{\partial y} \right]$$

$$= \frac{1}{\rho C_p} \left[k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} \right]$$

if k_x and k_y are constant. This is the Diffusion Equation.

Temperature changes cause stress changes according to the equation

$$\Delta\sigma_{ij} = - \delta_{ij}\, 3K\alpha\, \Delta T \qquad \text{(H-3)}$$

where $\Delta\sigma_{ij}$ = change in stress ij,

δ_{ij} = Dirac delta function,

K = bulk modulus,

α = linear thermal expansion coefficient, and

ΔT = temperature change.

The mechanical changes can also cause temperature changes as energy is dissipated in the system. This coupling is not modeled in FLAC because the heat produced is usually negligible.

H3. THERMAL-MECHANICAL INSTRUCTIONS

No new commands are needed to run thermal problems. However, new keywords are used on commands in the standard mechanical code. The user is strongly encouraged to gain familiarity with the mechanical logic berore attempting to model coupled problems.

List of Thermal Input Commands

APPly keyword value1 value2 <range>

This command is used to apply thermal boundary conditions and sources in the material. The keywords which can be used and the properties associated with value1 and value2 are listed below.

Keyword

Convection

Value1	Value2
temperature of medium to which convection occurs	convective heat transfer coefficient ($W/m^2°C$)

Flux

Value1	Value2
initial strength ($watts/m^2$)	decay constant (s^{-1})

APPly (continued)

Keyword

Radiation

Value1	Value2
radiative heat transfer coefficient (for black bodies,, this is the Stefan-Boltzmann constant, 5.668×10^{-8} $W/m^2 K^4$)	temperature of medium to which radiation occurs

Source

Value1	Value2
initial strength (watts/m depth)	decay constant (s^{-1})

Flux conditions can be applied over one row or column of gridpoints within the body or along a boundary. This feature enables the simulation of an internal line source, as a flux along an internal row or column of gridpoints. The **SOURCE** command results in a volume source of the specified strength in each zone of the specified range.

APPly (continued)

The decay constant in the SOURCE and FLUX options is defined by the equation

$$S_{curr} = S_{ini} * \exp[c_d(t_{curr} - t_{ini})] \qquad (H-4)$$

where S_{curr} = current strength,

S_{ini} = initial strength,

c_d = decay constant,

t_{curr} = current time, and

t_{ini} = initial time (when source was added)

To remove a FLUX or SOURCE condition, the condition should be applied with the strength replaced by s_{rep}, where

$$S_{rep} = -S_{ini} * \exp[c_d(t_{curr}-t_{ini})]$$

Fix T <value> <range>

The temperature is fixed at all points in <range>. If a <value> is given, the temperature is fixed at that value.

FREe T <range>

The temperature at points in <range> is allowed to change.

His **TEMPerature**

Temperature histories have been added.

Initial keyword = value <range>

TEMPERATURE has been added to the list of keywords. The temperature is initialized to the given value at all gridpoints and zones in the range specified.

MODEL keyword <range>

This command associates a thermal model with an area of the grid. The models available are as follows.

TH_Anisotropic

anisotropic conduction

TH_Isotropic

isotropic conduction

TH_General

isotropic conduction with thermal conductivity of the form

$$k(T) = k_1 + k_2 T^n$$

TH_Null

Zone is null. (Null zones model excavated material and insulators.)

PLot keyword <switch<=v>...>
<keyword <switch...>...>

TEMPERATURE has been added to the list of keywords.

Print keyword <keyword>

APPLY	Thermal boundary conditions and sources are also listed now.

The following main grid keywords have also been added.

CONductivity	thermal conductivity
SPec_heat	specific heat
Temperature	
THEXp	linear thermal expansion coefficient
XConduct	thermal conductivity in x-direction
YConduct	thermal conductivity in y-direction
CON1 CON2 N	parameters for general model

PROp keyword=value <keyword=value...> <range>

Properties can be assigned for models identified by the **MODEL** command. The property keywords and their meanings are given below.

Keyword	Description
SPec_heat	specific heat
Thexp	coefficient of linear thermal expansion

For the isotropic model:

CONductivity	thermal conductivity

For the anisotropic model:

XConductivity	thermal conductivity in x-direction
YConductivity	thermal conductivity in y-direction

For the general model:

CON1	k_1
CON2	k_2
N	n

SET keyword=value

Implicit (ON/OFF)

The implicit thermal calculation scheme is selected with this keyword.

MECH = value

The number of mechanical stpes done at each step is selected. (The use of this command is discussed in Section H4.)

NTHER = value

The number of thermal steps done at each step is selected. (The use of this command is discussed in Section H4.)

THDT = value

The thermal timestep is set to value.

NOTE: The program calculates the thermal timestep automatically for the explicit solution scheme. This keyword allows the user to choose a different timestep. If the program determines that the chosen step is too large, it will automatically reduce it to a suitable value when thermal steps

SET (continued)

are taken. It will not revert to a user-selected value until another **SET THDT** command is issued. The program selects a value which is usually just less than the critical value for numerical stability.

Thermal (ON/OFF)

If ON is selected, thermal steps are done when the **SOLVE** command is issued. If OFF is selected, thermal steps are not done.

Solve <keyword <= value>...>

The **SOLVE** command executes both mechanical and thermal timesteps. Calculation is performed until some limiting condition is reached. The limiting conditions may be the temperature increase at a point, the number of steps, the run time, or the simulated age. The limits are changed by the optional keywords listed below.

Age = A	problem time (in days)
Clock = t	run time (in minutes)
Implicit	implicit calculation scheme used. This is the same as using a **SET IMPLICIT** command prior to the **SOLVE** command.
Noage	turns off the previously requested test for exceeding age A
Step = s	timesteps
Temp = T	total temperature increase since the previous mechanical cycles

Solve (continued)

Defaults for these keywords are

T = 20 degrees
s = 500 timesteps
t = 5 minutes

The default for the age parameter is that the age is not tested until an age has been explicitly requested via an 'age = value' following a **SOLVE** command.

Note that T depends on the units adopted for temperature. Old limits apply when set or on restart, but they are reset to defaults when a **NEW** command is given. A time limit of greater than 24 hours (1440 minutes) will not be accepted. When a **SOLVE** command has been completed, the program will indicate which parameter has caused it to terminate. To ensure that it stops for the correct one, the values of the others should be set very high.

See Section H4 for information on enabling and disabling mechanical and/or thermal steps.

H4. SOLVING THERMAL-MECHANICAL PROBLEMS

FLAC has the capability to solve both transient and steady-state thermal and thermo-mechanical problems. Both implicit and explicit solution methods are available. If FLAC is used to solve thermal problems, the **CONFIG** command must be given with the THERMAL keyword before the grid is specified.

The thermal model must be specified for any zones which conduct heat (usually, the entire grid). If zones are made null mechanically, the thermal model automatically is made null as well. Usually, this is what is desired, but there are cases where mechanically null zones are required to conduct heat (e.g., if the thermal boundary has to be much further away than the mechanical boundary). In this case, the thermal model can be specified <u>after</u> the mechanical null model has been specified, thus reactivating the zones thermally. Although the zones are active, they will not be shown when printing or plotting unless they are given a mechanical model temporar-ily.

The most common way of using FLAC to solve thermomechanical problems is to come to initial mechanical equilibrium and then take thermal steps to a time of interest. Remember that transient thermal problems involve time (e.g., the solution may be required after 10 years of heating). AT this point, the mechanical problem has not been solved, although temperatures have been calculated. Mechanical steps are then taken until equilibrium is reached. This process is illustrated in Fig. H-1.

1. SETUP

. define grid, deform to desired shape
. define material models and properties
. define thermal models and properties
. set boundary conditions (thermal & mechanical)
. set initial conditions (thermal & mechanical)
. set any internal conditions, such as heat sources

2. STEP TO EQUILIBRATE MECHANICALLY

3. PERFORM ANY DESIRED ALTERATIONS such as excavations

4. STEP TO EQUILIBRATE MECHANICALLY

REPEAT steps 3 and 4 until "initial" mechanical state is reached for thermal analysis.

5. TAKE THERMAL TIMESTEPS until desired time is reached.

6. STEP TO EQUILIBRATE MECHANICALLY

REPEAT steps 5 and 6 until sufficient time has been simulated.

REPEAT steps 3 to 6 as necessary.

Fig. H-1 General Solution Procedure for Thermal-Mechanical Analysis

The **SOLVE** or **STEP** command is used to control both mechanical and thermal steps. When a thermal problem is started, FLAC automatically configures itself to run both thermal and mechanical steps when a **SOLVE** or **STEP** command is issued. To prevent one or another from being run, as is required to follow the procedure in Fig. H-1, use the commands

SET THERM ON/OFF

and

SET MECH ON/OFF

An alternative way of using the code is to solve both thermal and mechanical parts together. This is generally less efficient and is probably unnecessary, unless the user feels it is essential to accurately incorporate the thermal and mechanical loading simultaneously. One case in which such a procedure is necessary is probably when the WIPP creep law is used (with the creep version of FLAC). Because of the strong temperature-dependence of this law, it is necessary to have the correct temperature available for each mechanical step.

If the problem being run is such that mechanical equilibrium must be maintained at every step, it may be necessary to do more than one mechanical step for each thermal step. FLAC, by default, will only do one of each on each cycle. To do more than one, the **SET NMECH=value** command can be used. For example, **SET NMECH=10** will cause 10 mechanical steps to be done for each thermal step. There will also be cases in which more

than one thermal step is required for each mechanical step. For example, if a creep model is being used, the creep (mechanical) timestep may be greater than the thermal timestep. Rather than reducing the mechanical timestep, it is more efficient to use the **SET NTHER=value** command to increase the number of thermal steps run for each mechanical step.

Another alternative is to use the implicit algorithm described below to ensure that the timesteps are equal. A similar reasoning should be applied to the use of the groundwater flow logic. In this case, the thermal and groundwater <u>times</u> must be matched even though the <u>timesteps</u> are different. The appropriate commands to do this are **SET NGW** and **SET NTHER**. The groundwater timestep is not user-controllable at present, nor is there an implicit scheme for the groundwater logic. When using both thermal and groundwater options, bear in mind that there is no direct coupling between them—i.e., heat transfer by groundwater movement is not considered, nor is thermal expansion of the water.

Thermal Implicit Logic in FLAC

The usual one-dimensional explicit finite difference scheme can be written

$$\frac{\rho C_p}{k} \cdot \frac{T_i(t + \Delta t) - T_i(t)}{\Delta t} = \frac{T_{i+1}(t) - 2T_i(t) + T_{i-1}(t)}{(\Delta x)^2} \qquad \text{(H-5)}$$

where i is the gridpoint considered, and i-1 and i+1 are the points to the left and right, respectively,

t is the current time,

Δt is the timestep, and

Δx is the distance between gridpoints.

An implicit method can be derived by replacing the right-hand side of Eq. (H-5) by the expression:

$$\frac{1}{2}\left[\frac{T_{i+1}(t+\Delta t) - 2T_i(t + \Delta t) + T_{i-1}(t + \Delta t)}{(\Delta x)^2} + \frac{T_{i-1}(t) - 2T_i(t) + T_{i-1}(t)}{(\Delta x)^2}\right]$$

This method, known as the Crank-Nicholson method, has the advantage that it is stable for all values of Δt, but has the disadvantage of being implicit. This means that the temperature change at any point depends on the temperature change at other points. (This can be seen by rewriting the implicit scheme as

$$\frac{\rho c p}{k \Delta t} \Delta T_i = \left[\frac{T_{i+1} + \frac{1}{2}\Delta T_{i+1} - 2(T_i + \frac{1}{2}\Delta T_i) + T_{i+1} + \frac{1}{2}\Delta T_{i+1}}{(\Delta x)^2} \right]$$

since $T_k(t + \Delta t) = T_k(t) + \Delta T_k$.)

The implicit method requires that a set of equations be solved at each timestep for the ΔT_is.

In matrix notation, the explicit method can be written as

$$\underline{\Delta T} = \underline{\underline{C}}\ \underline{T}$$

where $\underline{\underline{C}}$ is a coefficient matrix,

$\underline{T}$ is a vector of the temperatures, and

$\underline{\Delta T}$ is a vector of the temperature change.

The implicit scheme can be written as

$$\underline{\Delta T} = \underline{\underline{C}}(\underline{T} + \tfrac{1}{2}\underline{\Delta T})$$

which can be rewritten as

$$(\underline{\underline{I}} - \tfrac{1}{2}\underline{\underline{C}})\underline{\Delta T} = \underline{\underline{C}}\,\underline{T}$$

where we need to solve for $\underline{\Delta T}$ at each timestep.

The matrix

$$(\underline{\underline{I}} - \tfrac{1}{2}\,\underline{\underline{C}})$$

is diagonally dominant and sparse, because only neighboring points contribute non-zero values to $\underline{\underline{C}}$.

This set of equations is thus efficiently solved by an iterative scheme. For ease of implementation as a simple extension of the explicit method, the Jacobi method is used. For the NxN system $\underline{A}\underline{x}=\underline{b}$, this can be written for the n^{th} iteration as

$$x_i(n+1) = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j\neq i}}^{N} \left[\frac{a_{ij}}{a_{ii}} x_j(n) \right] \qquad \begin{matrix} i = 1,2,\ldots N \\ n = 1,2,\ldots \end{matrix}$$

where a_{ij} are the array elements, and

b_i are the elements of the right-hand-side vector,

that is,

$$x_i^{(n+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{N} a_{ij} x_j^{(n)} \right] + x_i^{(n)}$$

In our case, this becomes

$$\Delta v_i^{(n+1)} = \frac{1}{(1+(1/2)C_{ii})} \left[\sum_{j=1}^{N} C_{ij} v_j - \sum_{j=1}^{N} (\delta_{ij} - \frac{1}{2} C_{ij}) \Delta v_j^{(n)} \right] + \Delta v_i^{(n)}$$

(H-6)

where C_{ij} are elements of the C array.

This equation shows the analogy between the implicit scheme and the explicit scheme which can be written as

$$\Delta T_i = \sum_{j=1}^{N} C_{ij} T_j$$

The amount of calculation required for each timestep is approximately n + 1 times that required for one timestep in the explicit scheme, where n is the number of iterations per timestep. This extra calculation can be more than offset by the much larger timestep permitted by the implicit method. However, the implicit scheme can give poor accuracy because it assumes that the temperature change is a linear function of time in a single timestep, which may not be accurate, especially when temperatures are changing fast, as they generally do near the beginning of a run.

Input Commands — The command sequence to use the implicit method is as follows:

```
SET THDT=value

SOLVE.....IMPLICIT
```

or

```
SET THDT=val n

SET IMPLICIT

STEP/SOLVE
```

The value of THDT is not restricted by numerical stability. The extra keyword on the **SOLVE** command switches to an implicit scheme. It is permissible to change between implicit and explicit solution methods at any time during a run, using the **SET IMPLICIT ON/OFF** commands.

Usage — The advantage of an implicit method is that the timestep is not restricted by numerical stability. The disadvantages are that:

(1) extra memory is required to use this method;

(2) a set of simultaneous equations must be solved at each timestep; and

(3) larger timesteps may introduce inaccuracy.

These disadvantages must be kept in mind when deciding which method to use. They are discussed below.

Memory Requirement — If an attempt is made to use the implicit method for a problem when the FLAC memory is almost full (typically when the **PRINT MEM** command reports at least 95% full), an error message may be generated. The only way to avoid this is to run a smaller problem or use the explicit method.

Solving a Set of Equations — The set of equations to be solved at each timestep is solved iteratively. Each iteration of the solution takes about the same length of time as a single step of the explicit method. The number of iterations depends on the timestep chosen and the particular problem being solved, but is always at least 3. Thus, the implicit scheme only offers an advantage over the explicit scheme if the timestep is much larger than that which the explicit scheme would use. On the other hand, the iterative scheme does introduce some restriction on the timestep. In general, a timestep between 100 and 10,000 times that used by the explicit scheme is satisfactory.

The program displays the iteration counter and a measure of convergence (the residual) to the left of the timestep counter while the implicit scheme is running. The user should check that the number of iterations being taken is such that the implicit scheme is indeed more efficient than the explicit scheme. If not, switch to the explicit scheme or change the timestep. This counter will also indicate if the method is not converging. If the residual is increasing with successive iterations, the method is not converging, and a smaller timestep must be used.

Inaccuracy Due to Large Timesteps — In the initial period of a solution, temperatures generally change much faster than later. In addition, the implicit scheme uses more iterations when modeling rapid changes. It therefore is appropriate to use a smaller timestep or, more likely, the explicit method, initially, and then

switch to the implicit method with a large timestep later in the solution. Convergence of the solution generally occurs in fewer iterations at later timesteps.

Selecting the Implicit Method — From the above discussion, it can be seen that the implicit method is most efficient when used at late times in the solution, and only if the timestep can be increased significantly over the one used by the explicit scheme.

H5. EXAMPLE PROBLEM — A SIMPLIFIED MODEL OF A NUCLEAR WASTE REPOSITORY

We wish to determine the state of stress around a room excavation in a nuclear waste repository. The object of this example is to consider only the thermal effects—so gravity is ignored, as are in-situ stresses.

The material in which the excavation is made is assumed to be isotropic and linearly elastic. The thermal conductivity is assumed constant in all directions.

The heat source strength is approximated by the following equation (Hart et al., 1981):

$$Q(t) = 12.4 \exp(-3.5\times10^{-9}t) + 288.0 \exp(-7.1\times10^{-10}t) + 10.1 \exp(-9.6\times10^{-11}t)$$

where Q is the thermal output in watts,
and

t is the time in seconds after emplacement.

The heat transfer at the tunnel wall is assumed to be by convection to air at a temperature of 27°C. The initial rock temperature is also 27°C.

For the FLAC simulation, a simple 10x20 square-zone grid was used. Symmetry conditions were imposed on the left- and right-hand sides as well as the top and bottom. This corresponds to the situation in Fig. H-2.

The symmetry conditions are modeled by adiabatic boundaries (the default thermal boundary). If the problem really consisted of just one excavation, it would have been necessary to put the outer boundaries further away from the source and opening. In that case, it would have been desirable to run the simulation twice --first, with adiabatic outer boundaries and, then, with constant temperature outer boundaries. The correct result would be between these two; only if the two results were close would the boundaries be far enough away.

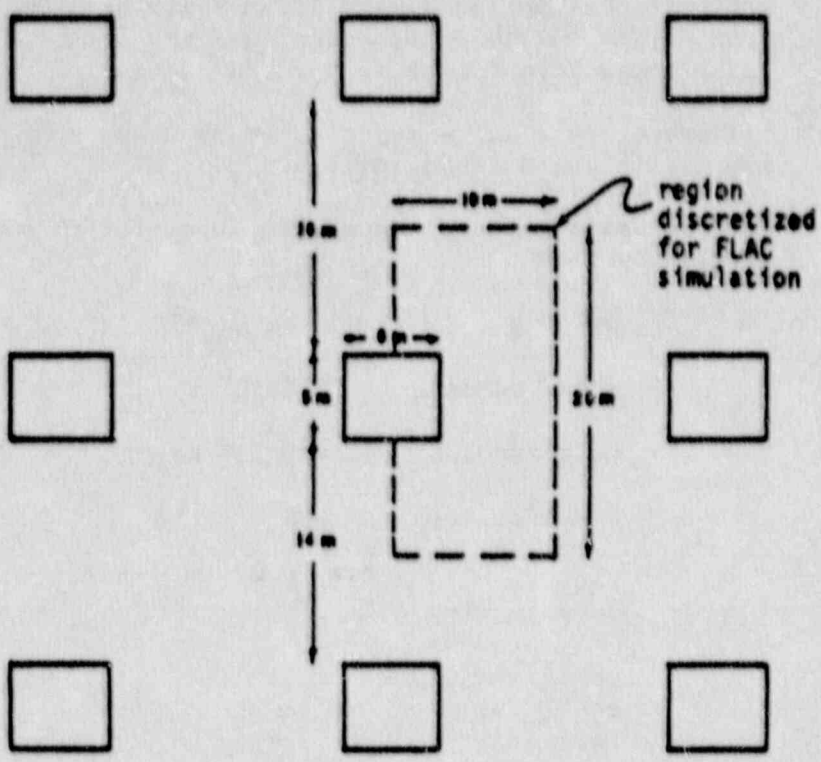


Fig. H-2 Series of Excavations Modeled

The mechanical boundary conditions for symmetry are that the top and bottom edges are prevented from moving in the y-direction, and the left and right edges cannot move in the x-direction.

The waste canister is represented by heat sources in two neighboring zones.

The rock mass material properties used for this simulation were

density	2200 kg/m^3
shear modulus	2.8x10^9 Pa
bulk modulus	4.7x10^9 Pa
specific heat	820 J/kg °C
thermal conductivity	2.1 W/m °C
coefficient of linear thermal expansion	1x10^{-5}/°C

The input is given in the following command sequence.

```
config th
gr 10,20
mod elas th iso
mod null i=1,3 j=8,12
prop bu=4.7e9 sh=2.8e9 dens=2200 cond=2.1 spec=820 thexp=1e-5
```

```
fix y j=1
fix y j=21
fix x i=1
fix x i=11
* initialize rock temperature
ini te=27
* apply heat source in zones 5,10; 6,10
app source 6.2 -3.5e-9 i=5,6 j=10
app source 144.0 -7.1e-10 i=5,6 j=10
app source 5.05 -9.6e-11 i=5,6 j=10
* apply convective boundary conditions
app convec 27 1  i=4 j=8,13
app convec 27 1  i=1,4 j=8
app convec 27 1  i=1,4 j=13
set clo 180
set ste 1500
set thermal off
solve  * without gravity, so there should be no out-of balance forces
tit
      Heat Source in Tunnel Wall
            ( Initial State )
save year0.sav
set thdt=8.64e4 * thermal timestep of 24
      hours
* solve thermal portion of problem
set mech off
set therm on
solve ste=365  tem=500 clo=180
* solve mechanical problem
set mech on
set therm off
solve
```

```
tit
      Heat Source in Tunnel Wall ( 1 Year )
save year1.sav
set therm on
set mech off
solve ste=365
set therm off
set mech on
solve
tit
      Heat Source in Tunnel Wall ( 2 Years )
save year2.sav
set therm on
set mech off
solve ste=1095
set therm off
set mech on
solve
tit
      Heat Source in Tunnel Wall ( 5 Years )
save Year5.sav
set therm on
set mech off
solve ste=1825
set therm off
set mech on
solve
tit
      Heat Source in Tunnel Wall
            ( 10 Years )
save Year10.sav
set therm on
set mech off
solve ste=3650
set therm off
set mech on
solve
```

```
tit
     Heat Source in Tunnel Wall
          ( 20 Years )
save Year20.sav
set therm on
set mech off
solve ste=10950
set therm off
set mech on
solve
tit
     Heat Source in Tunnel Wall
          ( 50 Years )
save Year50.sav
return
```

The saved states at times representing 1, 2, 5, 10 and 20 years can be restored to plot results such as those in Figs. H-3 through H-6.

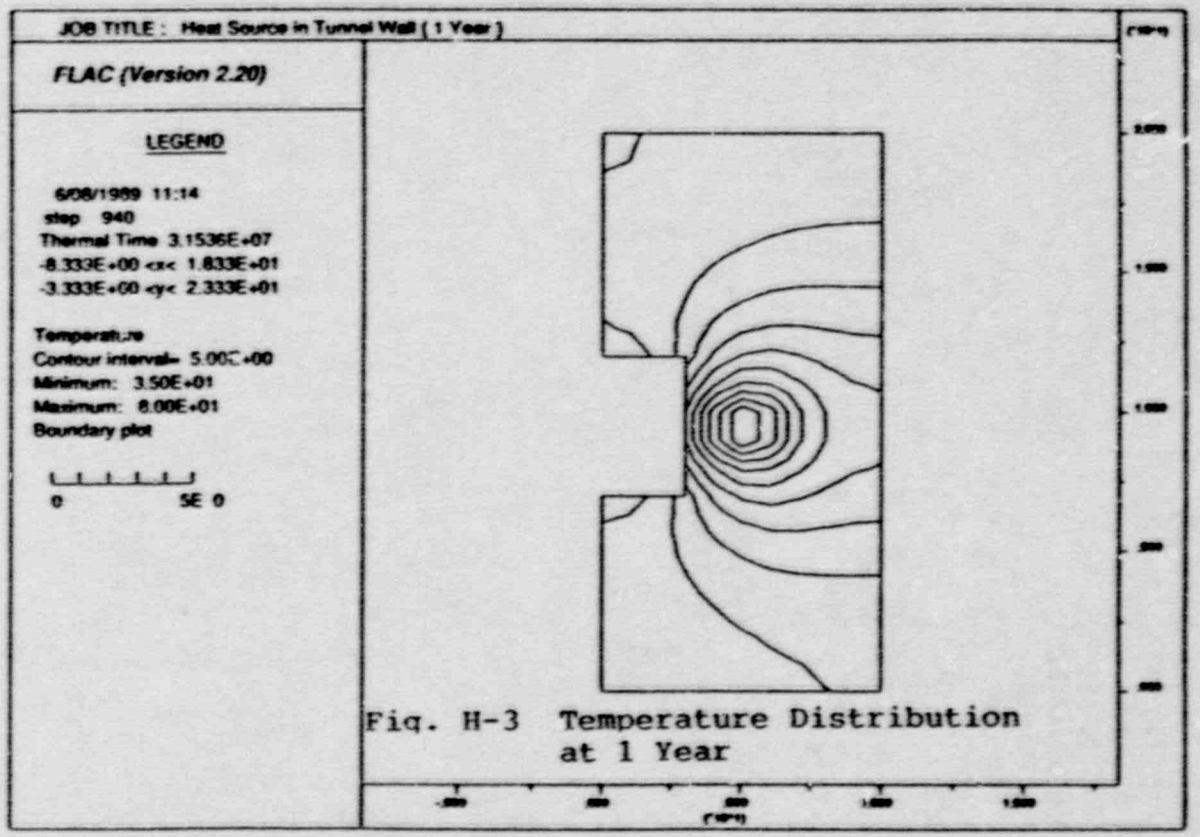


Fig. H-3 Temperature Distribution at 1 Year

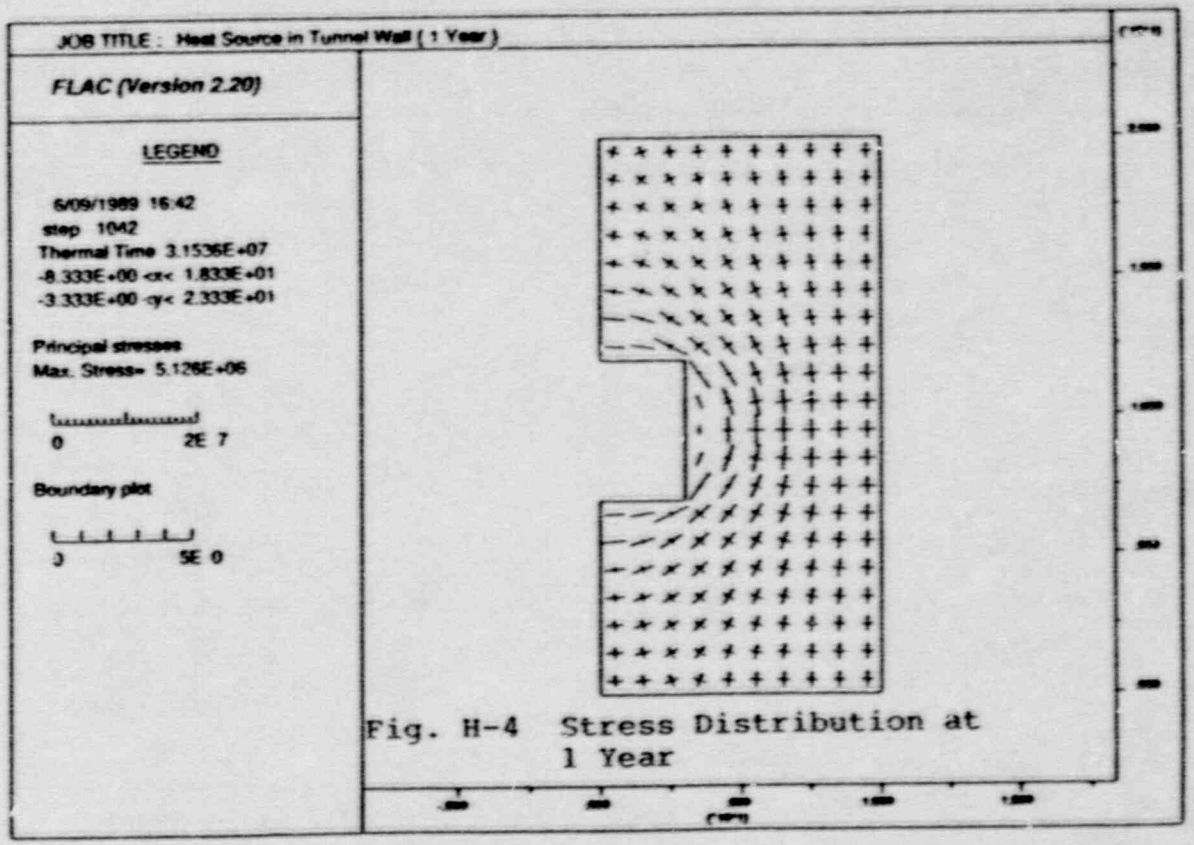


Fig. H-4 Stress Distribution at 1 Year

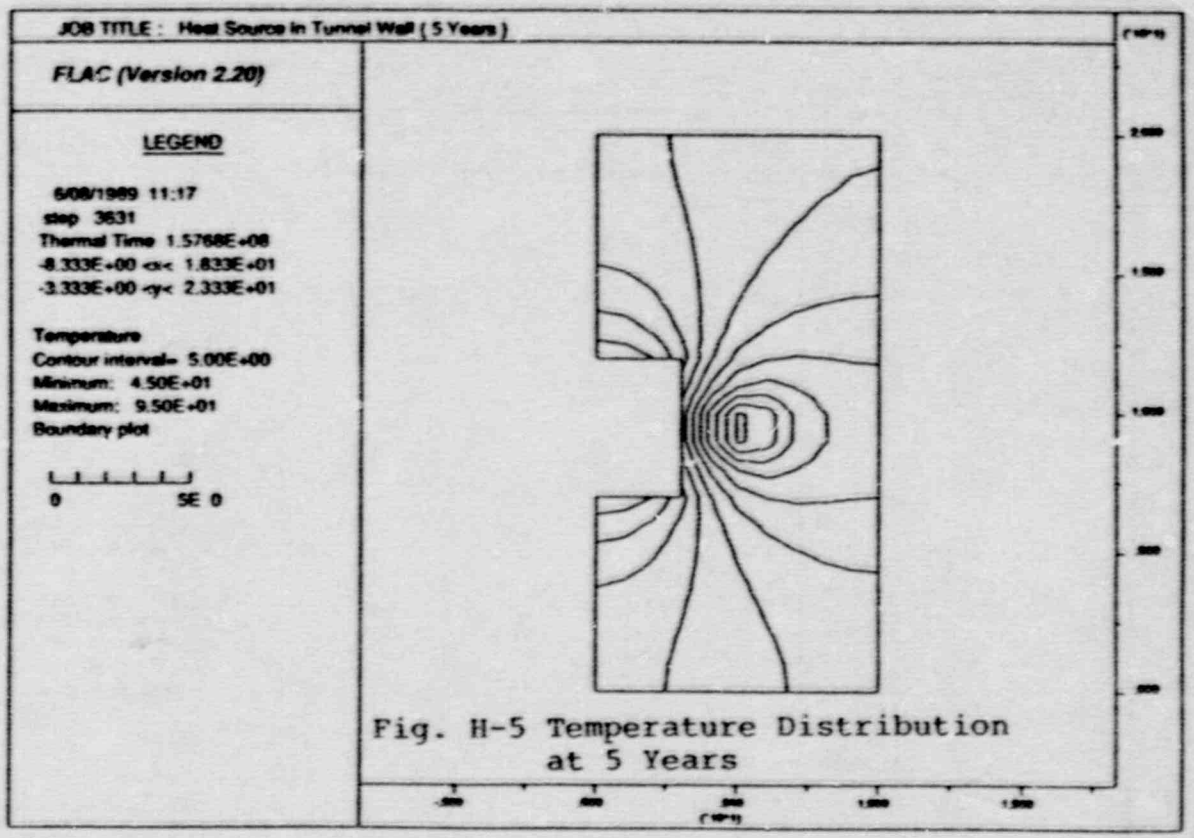


Fig. H-5 Temperature Distribution at 5 Years

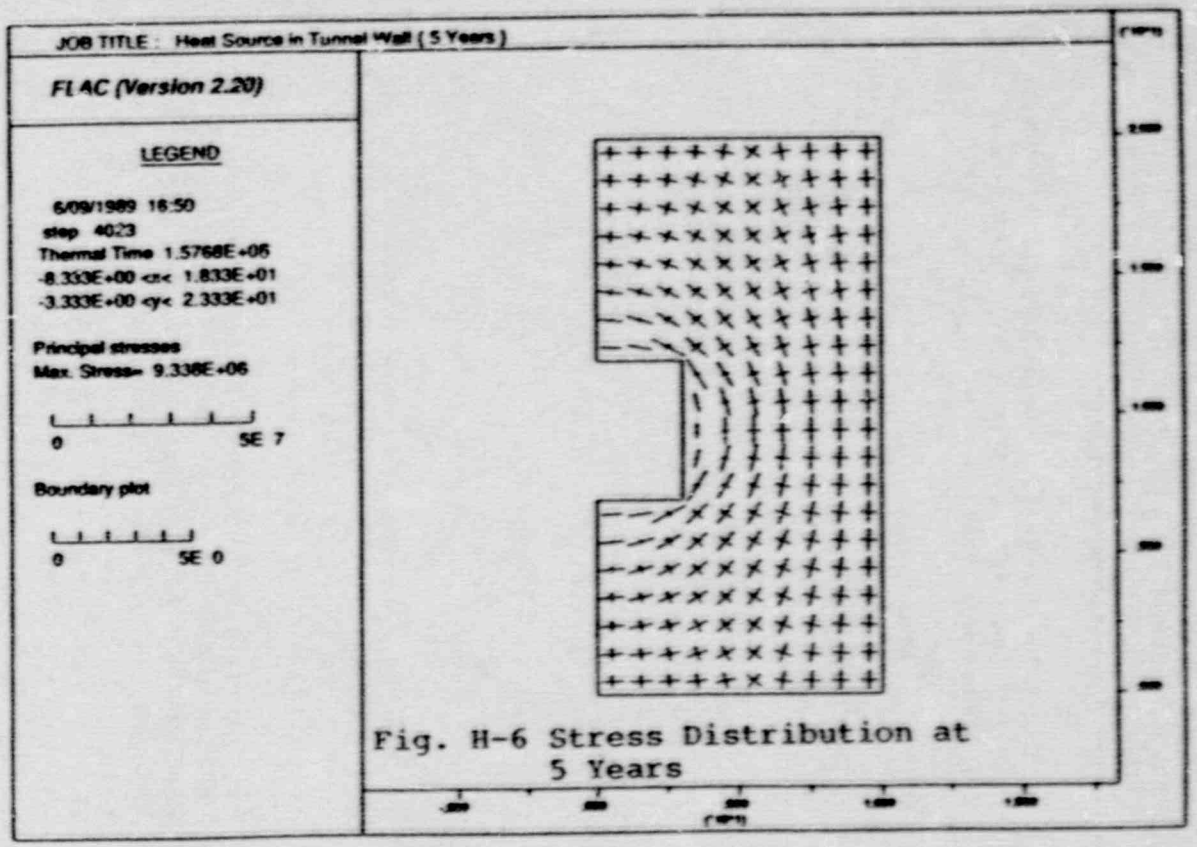


Fig. H-6 Stress Distribution at 5 Years

As an example, type

```
restore year1.sav
```

We now have the state after one year of simulation and can examine some of the conditions. Now, issue the command

```
print temp
```

to see the temperatures in the grid. Follow this by

```
plot temp  bound
```

or, for a hard copy, type

```
plot pen  temp  red  bound
```

Also, issue the command

```
plot stress  red  bound
```

to see the stresses developed around the opening. Recall that we had no initial stress or gravity so that the stresses shown are only those induced by the heat source.

We can restart from each of the saved states to see the changes as time progresses. Results at ages one and five years are shown in Figs. H-3 through H-6. The stress concentrations around the cavity are evident in Figs. H-4 and H-6.

REFERENCE

Hart, R. D., M. Christianson, W. Holman, and G. Hocking. "Numerical Modeling of Some Geotechnical Considerations Associated with Underground Isolation of Nuclear Wastes at the Savannah River Plant, South Carolina." Subcontract No. CNR3380 312, National Academy of Sciences, October 1981.

F L A C

A P P E N D I X I

CREEP MATERIAL MODELS

I1. INTRODUCTION

This option of FLAC can be used to simulate the behavior of materials which exhibit creep—i.e., time-dependent material behavior. Three creep models have been implemented in FLAC. These are:

(1) a classical visco-elastic model;

(2) a reference creep formulation for nuclear waste isolation studies; and

(3) a two-component power law.

The second model is commonly used in thermomechanical analyses associated with studies for the underground isolation of nuclear waste in salt, and the third can be used for mining applications. A description of these models and their implementation is provided in this appendix.

I2. DESCRIPTION OF CREEP CONSTITUTIVE MODELS

I2.1 Classical Visco-Elasticity (Kelvin Substance)

The classical description of Newtonian viscosity is that the rate of strain is proportional to stress. Stress-strain relationships can be developed for viscous flow in exactly the same way as those developed for the theory of elasticity. The derivation of the equations in three dimensions can be found, for example, in Jaeger (1969).

Visco-elastic materials exhibit both viscous and elastic behavior. One such material is the Kelvin material, which can be represented in one dimension by a spring and dashpot in parallel, as shown in Fig. I-1.

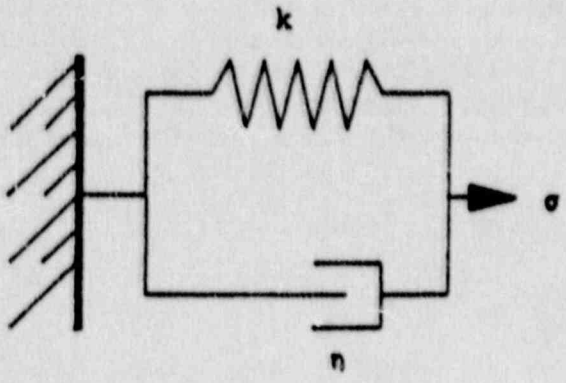


Fig. I-1 One-Dimensional Kelvin Model

The stress-strain law for this material can be written as

$$\sigma_{ij} = \sigma_{ij}^{(e)} + \sigma_{ij}^{(v)} \qquad \text{(I1)}$$

where $\sigma_{ij}^{(e)} = 2 G e_{ij}^{(d)} + K\delta_{ij} e_{kk}$,

$\sigma_{ij}^{(v)} = 2\eta \dot{e}_{ij}^{(d)}$,

G = shear modulus,

K = bulk modulus,

η = dynamic viscosity,

$e_{ij}^{(d)}$ = deviatoric strain = $e_{ij} - \frac{1}{3} e_{kk} \delta_{ij}$,

$\dot{e}_{ij}^{(d)}$ = deviatoric rate = $\dot{e}_{ij} - \frac{1}{3} \dot{e}_{kk} \delta_{ij}$,

e_{ij} = ij strain component,

$\sigma_{ij}^{(e)}$ = elastic part of stress, and

$\sigma_{ij}^{(v)}$ = viscous part of stress.

The material properties required for this model are the shear and bulk moduli (for the elastic behavior) and the viscosity.

I2.2 A Reference Creep Law for Nuclear Waste Isolation Studies

An empirical law has been developed (Senseny, 1985) to describe the time- and temperature-dependent creep of natural rock salt. In addition to the elastic component, the material is assumed to undergo creep, based on the equation

$$\frac{d\dot{\varepsilon}}{dt} = -\zeta\,(\dot{\varepsilon} - \dot{\varepsilon}_{ss})$$

— i.e.,

$$\dot{\varepsilon} = \dot{\varepsilon}_{ss} + e_a\,\zeta\,\exp(-\zeta t) \qquad (I2)$$

where $\dot{\varepsilon}$ = creep rate,

$\dot{\varepsilon}_{ss}$ = steady-state creep rate,

ζ = a rate parameter,

e_a = an integration constant, and

t = time.

Based on experiments, Senseny observed two regimes. For steady-state creep rates above a critical value $\dot{\varepsilon}^{*}_{ss}$, e_A is a constant $\dot{\varepsilon}_a$, and $\zeta = B\dot{\varepsilon}_{ss}$, where B is a constant.

For steady-state creep rates below $\dot{\varepsilon}^{*}_{ss}$,

$$e_A = \left[\frac{\dot{\varepsilon}_{ss}}{\dot{\varepsilon}^{*}_{ss}}\right]\dot{\varepsilon}_a \qquad \text{and} \qquad \zeta = B\,\dot{\varepsilon}^{*}_{ss}$$

Thus, Eq. (I2) can be written as

$$\dot{\varepsilon} = \begin{cases} \dot{\varepsilon}_{ss} + \dot{\varepsilon}_a B\dot{\varepsilon}_{ss}\exp(-B\dot{\varepsilon}_{ss}t) & \dot{\varepsilon}_{ss} \geq \dot{\varepsilon}^{*}_{ss} \\ \dot{\varepsilon}_{ss} + \dot{\varepsilon}_a B\dot{\varepsilon}_{ss}\exp(-B\dot{\varepsilon}^{*}_{ss}t) & \dot{\varepsilon}_{ss} \leq \dot{\varepsilon}^{*}_{ss} \end{cases}$$

Also, the steady-state strain rate is assumed to be given by

$$\dot{\varepsilon}_{ss} = A\sigma^{n}\exp(-Q/RT) \qquad \text{(I3)}$$

where σ = applied stress,

A, n, Q = parameters of the model,

R = universal gas constant, and

T = temperature (in Kelvin).

This formulation of the creep law is known as the RE/SPEC baseline creep law.

Another formulation of this law, known as the WIPP reference creep law, can be written as

$$\dot{\varepsilon} = \dot{\varepsilon}_s + \dot{\varepsilon}_p \qquad (I4)$$

where

$$\dot{\varepsilon}_p = \begin{cases} (A - B\varepsilon_p)\ \dot{\varepsilon}_s & \dot{\varepsilon}_s \geq \dot{\varepsilon}^* \\ (A - B\ (\dot{\varepsilon}^*_s/\dot{\varepsilon}_s) - \dot{\varepsilon}_p)\ \dot{\varepsilon}_s & \dot{\varepsilon}^*_s < \dot{\varepsilon} \end{cases}$$

$$\dot{\varepsilon}_s = D(\bar{\sigma})^n \exp(-Q/RT)$$

where

ε_p = primary creep strain,

ε_s = secondary creep strain,

$\dot{\epsilon}$ = rate used to calculate the strain-rate components using

$$\dot{\epsilon}_{ij} = (3/2)^{1/2} \left[\frac{\overset{d}{\sigma}_{ij}}{\bar{\sigma}} \right] \dot{\epsilon}$$

n, A, B, D, Q = parameters of the model,

R = universal gas constant,

$\bar{\sigma}$ = deviatoric stress, calculated as

$$(3/2)^{1/2} \left[\overset{d}{\sigma}_{ij} \overset{d}{\sigma}_{ij} \right]^{1/2}$$

where $\overset{d}{\sigma}_{ij}$ = ij-component of deviatoric stress; and

T = temperature (in Kelvin).

The WIPP and RE/SPEC formulations are different expressions of the same law, using slightly different notations. The WIPP formulation has been implemented in FLAC because it is better-suited for implementation in explicit computer codes because the effect of temperature and stress histories is automatically built into the formulation. The RE/SPEC formulation is only valid for constant stresses and temperatures, although it

can be modified to account for stress and temperature histories. The relationship between the notations used in the two laws is given in Table I-1.

Table I-1

NOTATION FOR WIPP AND RE/SPEC FORMULATIONS

WIPP Notation	RE/SPEC Notation	Units	Typical Value
A	$B\mathcal{A}_a$	—	4.56
B	B	—	127
D	A	$Pa^{-n}\ s^{-1}$	5.79×10^{-36}
n	n	—	4.9
Q	Q	cal/mol	1200
R	R	cal/mol K	1.987
$\dot{\varepsilon}^*_{ss}$	$\dot{\varepsilon}^*_{ss}$	—	5.39×10^{-8}

I2.3 The Two-Component Power Law

The Norton power law (Norton, 1929) is commonly used to model the creep behavior of salt. The standard form of this law is:

$$\dot{\varepsilon}_{cr} = A\,\bar{\sigma}^{\,n} \qquad \text{(I5)}$$

where $\bar{\sigma} = \frac{(3)^{1/2}}{2}\,(\sigma^d_{ij}\,\sigma^d_{ij})^{1/2}$,

σ^d_{ij} = deviatoric part of σ_{ij}, and

$$\dot{\varepsilon}_{ij} = \frac{(3)^{1/2}}{2}\,\dot{\varepsilon}_{cr}\,(\sigma^d_{ij}/\bar{\sigma})$$

Usually, the amount of data available does not justify adding any more parameters to the creep law. There are cases, however, where it is justifiable to use a law based on multiple creep mechanisms. FLAC, therefore, includes an option to use a two-component law of the form

$$\dot{\varepsilon}_{cr} = \dot{\varepsilon}_1 + \dot{\varepsilon}_2 \qquad \text{(I6)}$$

where

$$\dot{\varepsilon}_1 = \begin{cases} A_1 \bar{\sigma}^{n_1} & \bar{\sigma} \geq \sigma_1^{ref} \\ 0 & \bar{\sigma} < \sigma_1^{ref} \end{cases}$$

$$\dot{\varepsilon}_2 = \begin{cases} A_2 \bar{\sigma}^{n_2} & \bar{\sigma} \leq \sigma_2^{ref} \\ 0 & \bar{\sigma} > \sigma_2^{ref} \end{cases}$$

With these two terms, several options, described below, are possible.

1. <u>The Default Option</u>

$$\sigma_1^{ref} = \sigma_2^{ref} = 0$$

$\bar{\sigma}$ is always positive, so this is the one-component law with

$$\dot{\varepsilon}_{cr} = A_1 \bar{\sigma}^{n_1}$$

2. Both Components Active

$$\sigma_1^{ref} = 0$$

$$\sigma_2^{ref} = \text{"large"}$$

$$\dot{\varepsilon}_{cr} = A_1 \bar{\sigma}^{n_1} + A_2 \bar{\sigma}^{n_2}$$

3. Different Law for Different Stress Regimes

(a) $\sigma_1^{ref} = \sigma_2^{ref} = \sigma^{ref} > 0$

$$\dot{\varepsilon}_{cr} = \begin{cases} A_2 \bar{\sigma}^{n_2} & \bar{\sigma} < \sigma_{ref} \\ A_1 \bar{\sigma}^{n_1} & \bar{\sigma} > \sigma_{ref} \end{cases}$$

(b) $\sigma_1^{ref} < \sigma_2^{ref}$

$$\dot{\varepsilon}_{cr} = \begin{cases} A_2 \bar{\sigma}^{n_2} & \bar{\sigma} < \sigma_1^{ref} \\ A_1 \bar{\sigma}^{n_1} + A_2 \bar{\sigma}^{n_2} & \sigma_1^{ref} < \bar{\sigma} < \sigma_2^{ref} \\ A_1 \bar{\sigma}^{n_1} & \bar{\sigma} > \sigma_2^{ref} \end{cases}$$

(c) $\sigma_2^{ref} > \sigma_1^{ref}$

NOTE: Do not use this option. It implies that creep occurs for $\bar{\sigma} < \sigma_1^{ref}$ and for $\bar{\sigma} > \bar{\sigma}_2^{ref}$, but not for $\sigma_1^{ref} < \bar{\sigma} < \sigma_2^{ref}$.

I3. INPUT INSTRUCTIONS

All commands have the same structure as those in the standard version of FLAC. No new commands are required, but additional keywords are used with existing commands. The new keywords for each command are described below.

Model keyword <range>

Power	two-component power law
Viscous	classical viscosity
Wipp	WIPP reference creep formulation

Plot The command **PLOT HIS quantity TIME** will plot the history as a function of time rather than timestep number.

Print <keyword> <keyword>

The following keywords have been added:

Main Grid Keywords

ACt_energy	Q
A_wipp	A
B_wipp	B
D_wipp	D
E_dot_star	$\dot{\epsilon}_{ss}^{*}$
GAs_c	R
N_wipp	n
VIscosity	η
a_1	A_1
n_1	n_1
a_2	A_2
n_2	n_2
rs1	σ_1^{ref}
rs2	σ_2^{ref}

Print Creep

prints out information parameters, including when timestep was changed.

PROp keyword=value

In addition to the bulk and shear moduli, the creep models require the following properties:

Keyword	Description	Model
ACt_energy	activation energy (Q)	WIPP
A_wipp	A	WIPP
B_wipp	B	WIPP
D_wipp	D	WIPP
E_dot_star	$\dot{\varepsilon}_{ss}^{*}$	WIPP
GAs_c	gas constant (R)	WIPP
N_wipp	n	WIPP
VIscosity	dynamic viscosity	classical
a_1	A_1	power
n_1	n_1	power
a_2	A_2	power
n_2	n_2	power
rs1	σ_1^{ref}	power
rs2	σ_2^{ref}	power

SET <keyword=value>

The timestep for the creep model is controlled using the following keywords. A description of the automatic timestep controller appears in Section I4.

CREEPTIME

Creep time is initialized. This is useful if creep is to be started at a time other than zero (default = 0).

DT = value

The timestep is set manually to value. Whenever the timestep changes, the velocities are changed, to accommodate the fact that the FLAC velocities represent distance per step (default=0.0).

DT=AUTO The timestep will change automatically, according to the value of the maximum unbalanced force. In this case the following keywords can also be used.

MAXDT = value

The maximum timestep allowed is set to value (default = 1e4).

SET (continued)

MINDT = value

The minimum timestep allowed is set to value (default = 1e2).

FOBL = value

The timestep will be increased if the maximum unbalanced force falls below this value (default = 1e4)

FOBU = value

The timestep will be decreased if the maximum unbalanced force goes above this value (default = 1e5)

LMUL = value

The timestep will be multiplied by value if the unbalanced force falls below FOBL. (LMUL should be greater than 1, default = 2.0)

UMUL = value

The timestep will be multiplied by value if the unbalanced force exceeds FOBU. (UMUL should be less than 1, default = 0.5)

SET (continued)

LATENCY = value

the minimum number of timesteps which must elapse between timestep changes (default = 100).

NOTE: Often, it is most stable to set UMUL=1, so the timestep can only increase, but never decrease. In this case, FOBU can be set very high, because it is not used.

SOLVE <keyword = value>

Two keywords have been added to control when timestepping is stopped. These are:

Age = A	problem time
Noage	turns off the previously-requested test for exceeding age A.

The age parameter is not tested until an age has been explicitly requested via an "age=value" following a **SOLVE** command.

I4. SOLVING CREEP PROBLEMS WITH FLAC

I4.1 Introduction

The major difference between creep and other constitutive models in FLAC is the concept of problem-time in the simulation. For creep runs, the problem-time and timestep represent real time, while in the other constitutive models the timestep is an artificial quantity, used only as a means of stepping to equilibrium. This also has an effect on the velocities—velocities in FLAC are actually measured in units of distance **per step** rather than distance per time. The creep models introduce an exception to this rule. Although, internally, FLAC continues to calculate distance per step, when velocities are printed, plotted, or initialized, or histories are taken, they are calculated as distance/time, unless the timestep is zero—in which case, units of distance/step are used, as in the standard models. The timestep, and how to control it in FLAC, are described below.

I4.2 Timesteps in FLAC

For time-dependent phenomena such as creep, FLAC allows the user to define a timestep. The default for this timestep is zero, in which case, the program treats the material as linearly elastic. This can be used to attain equilibrium before starting a creep simulation. The constitutive laws for creep make use of the timestep in their equations, so timestep may affect the response.

Although the user may set the timestep, it is not arbitrary. If a system is desired to always be in mechanical equilibrium (as in a creep simulation), the time-dependent stress changes produced by the constitutive law must not be large compared to the strain-dependent stress changes, otherwise out-of-balance forces will be large, and inertial effects (which are theoretically absent) may affect the solution. The timestep must therefore be chosen small enough that the maximum unbalanced force is smaller than some tolerance. FLAC allows the user to set the timestep manually, or allow the program to control it automatically. If the timestep is changed automatically, it can be decreased whenever the maximum unbalanced force exceeds some threshold, and increased whenever it goes below some other level. Typical out-of-balance force criteria for the problem being solved can be determined by observing the out-of-balance force which occurs near equilibrium in the initial stage of the problem when only elastic effects occur (i.e., with the timestep set to zero). It has been found that continuous adjustment of the timestep creates mechanical "noise" in the system, so a different scheme has been introduced. After a timestep change has occurred, there is a user-defined "latency period" (e.g., 100 steps) during which no further adjustments are made, allowing the system to settle. Normally, the timestep will start at a small value, to accommodate transients such as excavation, and then increase as the simulation proceeds. If a new transient is introduced, it may be desirable to reduce the timestep manually, and then let it increase again automatically

The **SET** command is used to set the timestep, and the parameters required to allow it to change automatically. The new keywords are listed in Section I2.

For the WIPP model, the creep rate is temperature-dependent. This model, therefore, should only be run with the thermal option of the code. The code does not check that the timesteps used for creep and thermal steps are consistent—this is the user's responsibility. Read Appendix H to find out how to keep the timesteps consistent.

I4.3 The Out-of-Plane Stress

When the creep laws are used, the out-of-plane stress (σ_{zz}) is calculated. Because the creep laws include σ_{zz} in the deviatoric stress calculation used to establish creep rates, the **CONFIG Z_D** command must be used, and values of σ_{zz} are significant. This stress component can be initialized with the **INITIAL** command and adjusts as creep occurs, in the same way as the other components.

I5. EXAMPLE PROBLEMS

I5.1 Example 1: Parallel Plate Viscometer -- Classical Model

Suppose that a material with viscosity η is steadily squeezed between two parallel plates which are moving at a constant velocity V_0. The two plates have length 2l and are a distance 2h apart. The material is prevented from slipping at the plates. The approximate analytical solution, given by Jaeger (1969), is

$$v_x = 3V_0x(h^2 - y^2)/2h^3$$

$$v_y = V_0y(y^2 - 3h^2)/2h^3$$

$$\sigma_{xx} = 3\eta V_0[3(h^2 - y^2) + (x^2 - l^2)]/2h^3$$

$$\sigma_{yy} = 3\eta V_0(y^2 - h^2 + x^2 - l^2)/2h^3$$

$$\sigma_{xy} = -3(V_0\eta xy)/h^3$$

The problem is illustrated in Fig. I-2.

To solve the problem with FLAC, advantage can be taken of the symmetry about the x- and y- axes. The top right quadrant only need be modeled. For compatibility with the approximations of the analytical solution, artificial forces have to be applied at the "free" right-hand edge, and small strain logic is used.

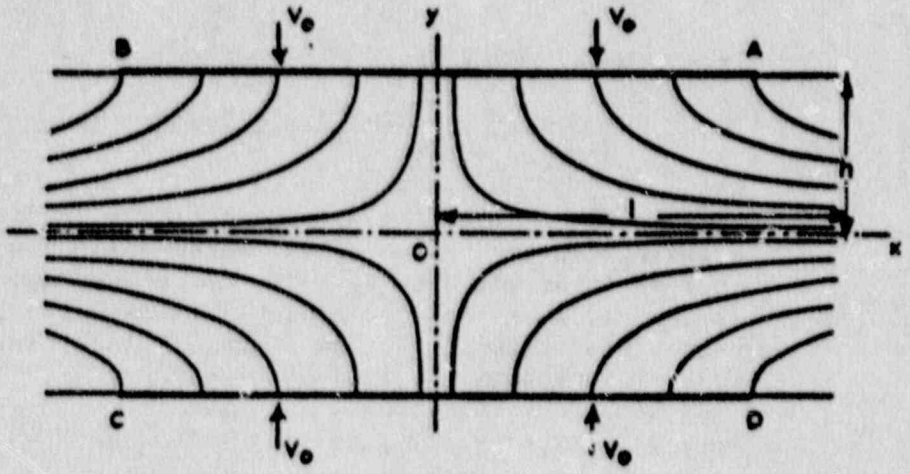


Fig. I-2 Parallel Plate Viscometer [Jaeger, 1969]

The material properties are:

density	1 kg/m^3
shear modulus	5×10^8 Pa
bulk modulus	1.5×10^9 Pa
viscosity	1×10^{10} kg/ms

The input is given in the following command sequence:

```
* classical viscosity - parallel plate test
conf z_d
gri 10 5
m vis
fix x i 1
fix y j 1
fix x y j 6
ini yv -1e-4 j 6
app xf 4.5e5 i 11 j 1
app xf 8.64e5 yf -2.4e5 i 11 j 2
app xf 7.56e5 yf -4.8e5 i 11 j 3
app xf 5.76e5 yf -7.2e5 i 11 j 4
app xf 3.24e5 yf -9.6e5 i 11 j 5
prop d 1 sh .5e9 bu 1.5e9 visc 1e10
set dt=1
his xv yv sx sy sxy sz i 4 j 4
ste 500
```

Figure I-3 is a plot of σ_{xx} contours. Other variables can be plotted and compared with the analytical solutions.

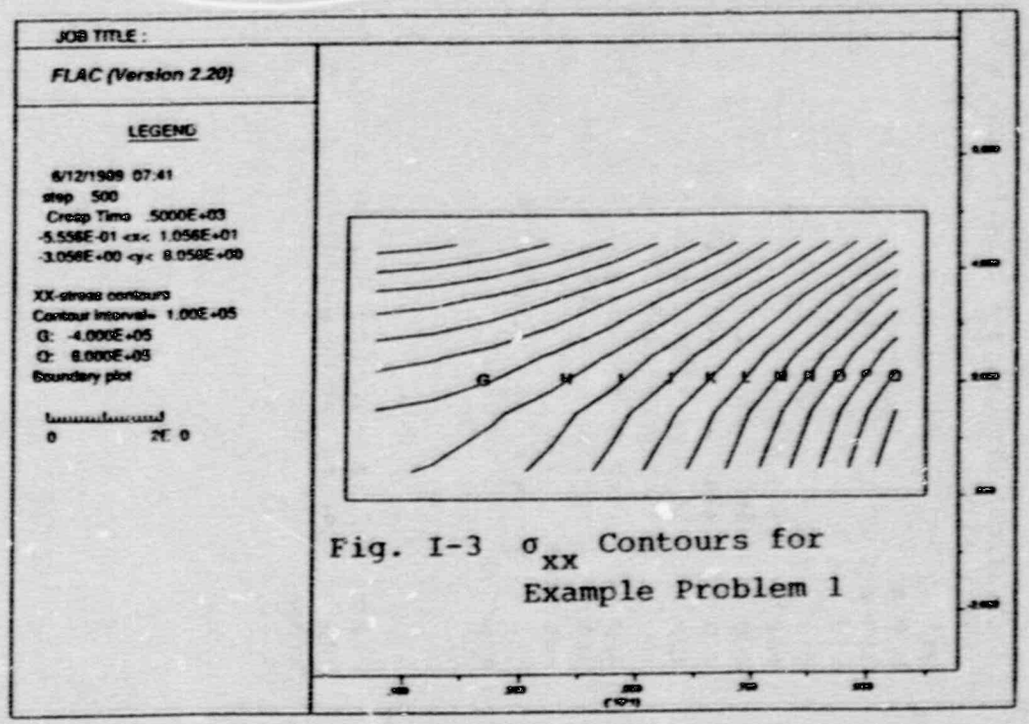


Fig. I-3 σ_{xx} Contours for Example Problem 1

I5.2 Example 2: Parallel Plate Viscometer—WIPP Model

The same problem can be modeled using the WIPP model. The following data file contains the commands necessary to run this problem. Note that, for this problem, it is essential to have the temperatures in the grid available, because they are used by the WIPP creep law. In this case, the **INI TEM** command is used to input a uniform temperature of 300K.

```
* parallel plate test WIPP model
config z th
g 10 5
m wipp
title
parallel-plate viscometer with WIPP model
fix x i 1
fix y j 1
fix x y j 6
ini yv -1e-5 j 6
ini tem 300
appl xf 4.5e5 i 11 j 1
appl xf 8.64e5 yf -2.4e5 i 11 j 2
appl xf 7.56e5 yf -4.8e5 i 11 j 3
appl xf 5.76e5 yf -7.2e5 i 11 j 4
appl xf 3.24e5 yf -9.6e5 i 11 j 5
prop d 2600 sh 12.4e9 bu 20.7e9
prop gas 1.987 act 12e3 n_wipp 4.9 D_wipp 5.79e-36
prop a_wip 4.56 b_wip 127 e_dot 5.39e-8
set dt 1e4
wind -1 15 -4 8
ste 900
ret
```

I5.3 Example 3: Hole in a Plate-Power Law

The power law in FLAC is used to solve the problem of a circular hole in a flat plate. A comparison is made with an analytical solution.

I5.3.1 Problem Statement — A flat square plate with a small circular hole in the center is illustrated in Fig. I-4. The plate is subjected to a constant pressure on all sides, and is under conditions of plane strain. The creep behavior of the plate material is defined by a single component power law — i.e.,

$$\dot{\varepsilon}_{cr} = A \bar{\sigma}^{n} \qquad \text{(I-5)}$$

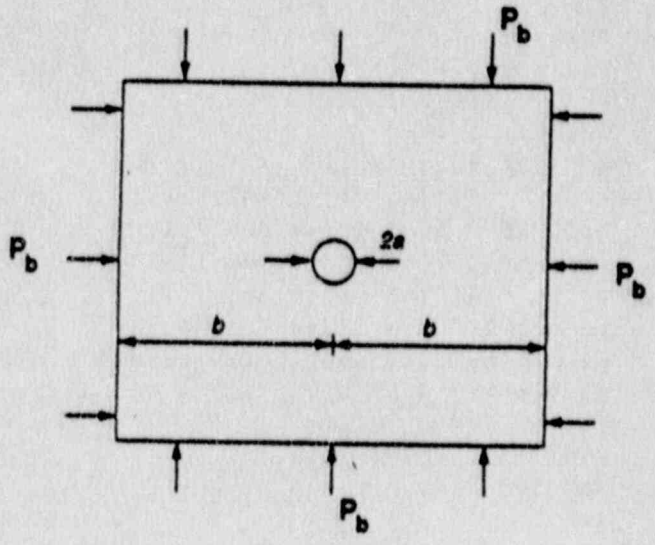


Fig. I-4 Flat Plate with Circular Hole

For this problem, $A = 1 \times 10^{-7}$ MPa^{-3} yr^{-1} and $n = 3$. The elastic properties of the plate are $E = 820$ MPa and $\upsilon = 0.3636$

An analytical steady-state solution to this problem (for a circular plate) has been provided by van Sambeek (1986), and is reproduced below:

$$\sigma_R = - P_b \left[\frac{1 - (a/r)^{2/n}}{1 - (a/b)^{2/n}} \right]$$

$$\sigma_\theta = - P_b \left[\frac{1 - [(n-2)/n] \ (a/r)^{2/n}}{1 - (a/b)^{2/n}} \right]$$

$$\sigma_z = \frac{\sigma_R + \sigma_\theta}{2} \qquad \text{(I-6)}$$

$$\dot{u}_R(a) = - A \frac{(3)^{1/2}}{2} \left[\frac{(3)^{1/2}}{n} P_b \frac{1}{1 - (a/b)^{2/n}} \right]^n a$$

where σ_R, σ_θ are radial and tangential stress components,

P_b is the applied boundary stress,

σ_z is the out-of-plane stress component,

$\dot{u}_r$ is the radial displacement rate,

a,b are the radius of hole and half-width of plate, respectively, and

r = radius to point of calculation.

I5.3.2 FLAC Solution — FLAC was used to solve this problem with a relatively coarse grid with the plate half-width (b) = 400 times hole radius (a).

One quarter of the plate in Fig. I-4 was modeled with FLAC, as shown in Fig. I-5. A pressure of 100 MPa was applied to the top and right boundaries, the bottom was restrained in the vertical direction, and the left boundary was restrained in the horizontal direction. The last two conditions are required to represent the symmetry correctly. The initial stresses, corresponding to an elastic plate in plane strain without a hole, were:

$$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = P_b = 100 \text{ MPa}$$

The plate was allowed to come to elastic equilibrium by setting the creep timestep to zero. Then, the creep timestep was set and allowed to increase automatically until steady state was reached. The data file for this problem is given below.

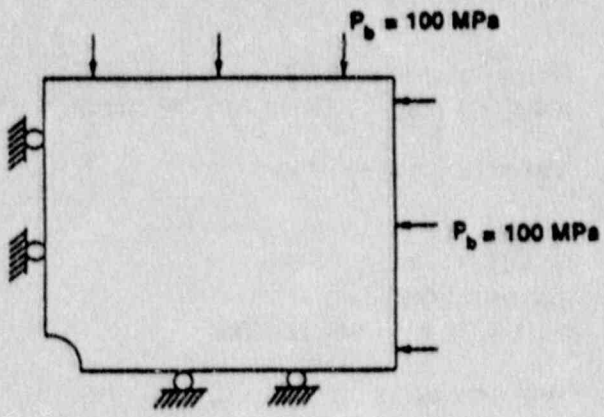
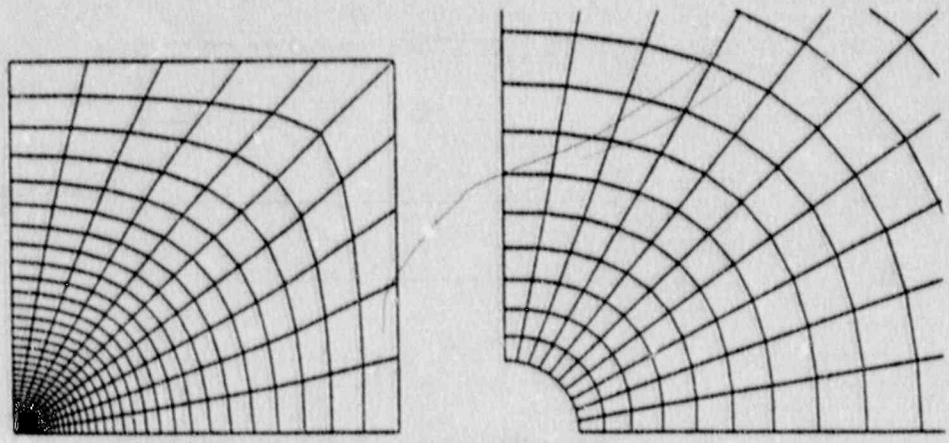


Fig. I-5 FLAC Representation of a Flat Plate

```
conf z d
gr 10,50
m pow
gen 0,1 0,400 400,400 1,1 rat 1 1.1 i 1 6
gen same same 400,0 1,0 rat 1 1.1 i 6,11
ini x=1 y=0 i=11 j=1
ini x=0.987688 y=0.156434 i=10 j=1
ini x=0.951057 y=0.309017 i=9 j=1
ini x=0.891007 y=0.45399 i=8 j=1
ini x=0.809017 y=0.587785 i=7 j=1
ini x=0.707107 y=0.707107 i=6 j=1
ini x=0.587785 y=0.809017 i=5 j=1
ini x=0.45399 y=0.891007 i=4 j=1
ini x=0.309017 y=0.951057 i=3 j=1
ini x=0.156434 y=0.987688 i=2 j=1
ini x=0 y=1 i=1 j=1
gen adj
gen adj
gen adj
```

```
* Properties and stresses in Pascal units (not MPa)
prop a_1=1e-25 n_1=3 bulk=1e9 shear=3e8 dens=2000
*
*boundary and initial conditions
*
fix x i=1
fix y i=11
apply press 100e6 j=51
ini sxx -100e6 syy -100e6 szz -100e6
*
* keep some histories
*
his unbal
his xvel i=11 j=1
* come to elastic equilibrium
ste 1000
sav hol1.sav
his reset
his n=100
his unbal
his xvel i 11 j 1
his sxx syy szz i 10 j 9
his sxx i 10 j 15
his sxx i 10 j 23
his sxx i 10 j 30
*
* Set creep parameters
* Creep parameters based on elastic out-of-balance forces
* Timestep will double when fob less than 5e3, until dt=0.1
set dt=1e-4
set fobl=5e3 fobu=1e8 imul=2 umul=1 maxdt=1e-1 mindt=0
set dt=auto
ste 5000
sav hol2.sav
set maxdt=1
ste 4000
sav hol3.sav
```

I5.3.3 Results — A relatively course grid, consisting of only 500 zones, was used for this problem (Fig. I-6). The results for this case are summarized in Figs. I-7 to I-10. Figure I-7 shows the radial velocity (i.e., u_R) history of a point on the circumference of the hole. It is apparent from this figure that the velocity starts off significantly higher than the steady-state solution, but drops rapidly to a fairly steady value a little (3%) below the analytical solution. The initial high value is expected because the pre-creep state is far from equilibrium. The stress components obtained from FLAC are compared with the analytical solution in Figs. I-8, I-9 and I-10. It is obvious from these figures that the FLAC results are virtually identical to the analytical solution.



(a) (b)

Fig. I-6 (a) FLAC Grid; (b) Close-Up View of Grid Around Hole

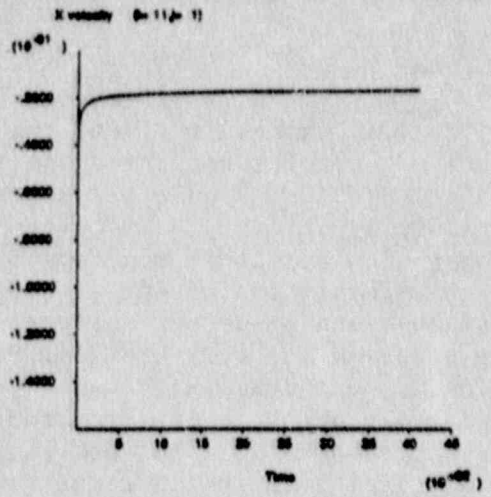


Fig. I-7 Radial Velocity at Borehole Edge vs Time

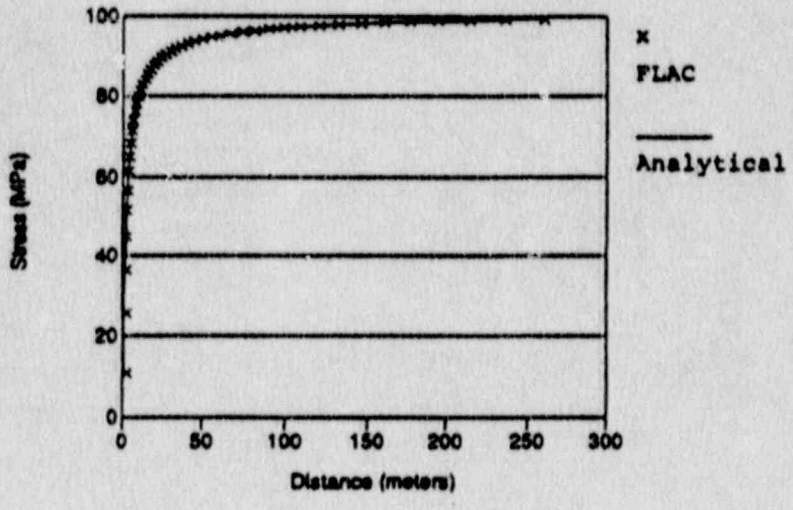


Fig. I-8 Steady-State Radial Stress (σ_R) vs Distance from Borehole

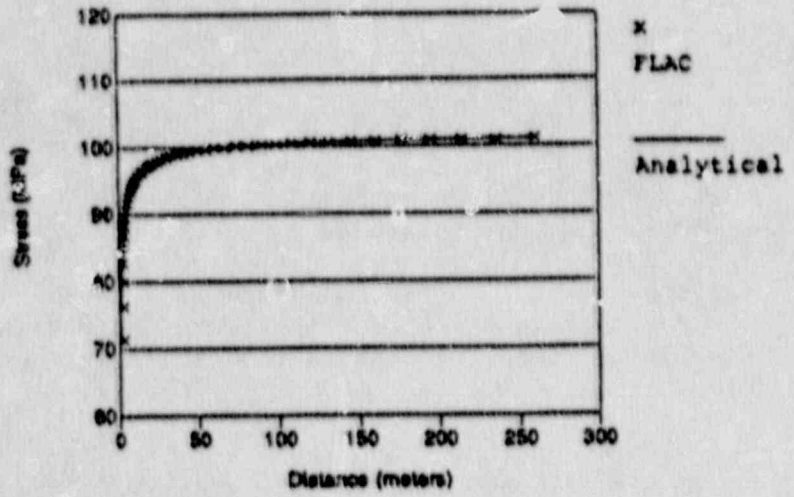


Fig. I-9 Steady-State Hoop-Stress (σ_θ) vs Distance from Borehole

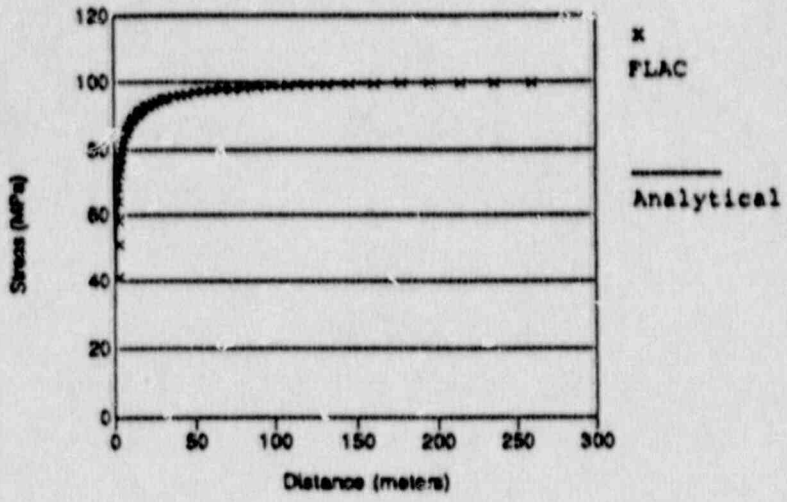


Fig. I-10 Steady-State Out-of-Plane Stress (σ_{zz}) vs Distance from Borehole

I6. REFERENCES

Jaeger, J. C. Elasticity, Fracture and Flow (3rd Ed.). New York: John Wiley & Sons, Inc., 1969.

Norton, F. H. Creep of Steel at High Temperatures. New York: McGraw-Hill Book Company, 1929.

Senseny, Paul, E. "Determination of a Constitutive Law for Salt at Elevated Temperature and Pressure," American Society for Testing and Materials Reprint 869, 1985.

Van Sambeek, L. L. Creep of Rock Salt Under Inhomogeneous Stress Conditions. Ph.D. Thesis, Colorado School of Mines, Golden, Colorado, 1986.

FLAC
QUICK REFERENCE COMMAND LIST
(Includes Optional Modules)

COMMAND	POSSIBLE KEYWORDS*
Apply	Pressure Xforce, Yforce, SXx,SYy,SXy Well, Discharge [Convection, Radiation, Flux, Source]* <range, long,short,both from ... to ...>
ATtach	Aside <long> from . . . to . . . / Bside <long> from . . . to . . .
Call	filename
Config	Axi, Gw, P-stress [Therm]*
Cycle	n
Fix	X, Y, P [T]* <Mark> <range>
FRee	X, Y, P [T]* <Mark> <range>

*Terms in brackets < > are optional; <range> means <i=__, j=__>.

*thermal option only

COMMAND	POSSIBLE KEYWORDS
GEn	x1,y1 x2,y2 x3,y3 x4,y4 <RAT xrat,yrat> <range>
	ARC xc,yc xb,yb ang CIRCLE xc,yc rad LINE xb,yb xe,ye TABLE n ADJUST
Grid	icol, jrow
HELP	
His	NSTE = n Dump, List, Write, Reset
His	any main grid variable, angle, sig1,sig2, unbal
Initial	Pp, X, Y, SXx, SYy, SXY, XDisp, XVel, YVel [T]* <Mark> <region> <ADD> <MUL>
INTerface	See **INTERFACE** command, Appendix D for more information.

COMMAND	POSSIBLE KEYWORDS
MArk	<range>
Model	Anisotropic Elastic, Mohr-Coulomb, Null, Ubiquitous, SS <REGion=i,j> <range> [Viscous, Wipp, Power]** [th_iso, th_aniso, th_general]*
New	
PLot	Boundary, Disp, Grid, Pen, STress, WAter, Velocity, PP, SXx, SYy, SXY, XDisp, Ydisp, XVel, YVel, STAte, BEam, Rf, Cable, MODel, Region, E_p, His n, SIG1, SIG2, Hoek, Mohr, Plas, Tens, Cforce, Apply, Theta, Num, SZZ, GNum, Mark, Flow, Head, Table n, Attach, Fix, any model variable <Color*, Interval, Max, Zero, Line, Fill, Noh, Scales>

**creep option only

*Color depends on graphics mode and palette

COMMAND	POSSIBLE KEYWORDS
Print	Apply, Attach, Interface, Limits, State, Tables, XReaction, YReaction, MArk, Mem, STRUc, SIG1, SIG2, Theta, Fix, Mark, any main grid variable, any property <range>
PROp	Bulk, Cohesion, Density, Dilation, Friction, Shear, Tension, Xmod, Ymod, NUYx, NUZx, JCohesion, JFriction, JAngle, JTens, Permeability, porosity, k11, k12, k22 <VAR = xvar,yvar>, FTable, Ctable, Dtable <REGion=i,j> <range> [creep model properties]**
Quit	
REstore	filename
HETurn	
SAve	filename
SCline	n (xb,yb),(xe,ye) <reset>

COMMAND	POSSIBLE KEYWORDS
SET	A3, A4, Asp, Back, BAud, Col, CGA, CSC, EGA, Force, Gravity, HBM, HBS, Large, Log, Mono, Output, Pal, Plot, PF, PSL, Small, Step, Time, UCS, Leg, VGA, Pltf, Pltc, Pltt, Echo, Cols, Flo, NMech, NGW, DAtum [thdt, implicit, nther, ther]*, [creep parameters]**,
SOlve	STep, Clock Force [Age, Noage, Implicit]*
STEp	n
STop	
STRucture	

STRuct [Beam / Cable] **Begin** [Grid i,j / Node n / x_b,y_b] **End** [Grid i,j / Node n / x_e,y_e] <Beg=n Prop=k Tens=t>

STRucture	Node=n keyword
STRucture	PROP=n E, I, Area, Kbond, SBond, Yield
SYS	dir, ren, copy, del, type

COMMAND	POSSIBLE KEYWORDS
Table	n x1,y1 x2,y2 x3,y3 . . .
Title	
Unmark	<range>
WAter	DEN, TAB, Bulk Unmark <range>
Window	xlo, xhi, ylo, yhi AUTO

NRC FORM 335 (2-89) NRCM 1102 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)

NUREG/CR-5430
Vol. 2

2. TITLE AND SUBTITLE

FLAC (Fast Lagrangian Analysis of Continua)
Version 2.20

User's Manual

3. DATE REPORT PUBLISHED

MONTH	YEAR
October	1989

4. FIN OR GRANT NUMBER

FIN D1016

5. AUTHOR(S)

Mark Board

6. TYPE OF REPORT

Formal

7. PERIOD COVERED (Inclusive Dates)

8. PERFORMING ORGANIZATION – NAME AND ADDRESS (If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.; if contractor, provide name and mailing address.)

Itasca Consulting Group, Inc.
1313 5th Street, SE, Suite 210
Minneapolis, MN 55414

9. SPONSORING ORGANIZATION – NAME AND ADDRESS (If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)

Division of High-Level Waste Management
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555

10. SUPPLEMENTARY NOTES

11. ABSTRACT (200 words or less)

FLAC, Version 2.20, is a two-dimensional, large-strain, explicit finite difference program for analysis of problems in geotechnical engineering. Principal features of the code include various mechanical constitutive models, heat transfer analysis, fluid flow analysis, structural element coupling and frictional and cohesive interfaces. The thermal and fluid flow analyses may be coupled to the mechanical portion of the code. The following report presents the documentation of FLAC in compliance with NUREG-0856, Documentation of Computer Codes for High Level Waste Management. The report is in three volumes: the first presents the mathematical formulation of the various portions of the code; the second, a user's manual; and the third, assessment of the code and support.

12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)

FLAC, Version 2.20
large-strain
finite difference
computer program
geotechnical
mechanical
thermal
fluid flow
coupled
interfaces
structural elements
NUREG-0856
documentation
code assessment

13. AVAILABILITY STATEMENT

Unlimited

14. SECURITY CLASSIFICATION

(This Page)
Unclassified

(This Report)
Unclassified

15. NUMBER OF PAGES

16. PRICE

*U.S. GOVERNMENT PRINTING OFFICE:1989-262-436:00295

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

SPECIAL FOURTH-CLASS RATE
POSTAGE & FEES PAID
USNRC
PERMIT No. G-67