1 INTRODUCTION

1.1 Overview

FLAC3D is a three-dimensional explicit finite-difference program for engineering mechanics computation. The basis for this program is the well-established numerical formulation used by our two-dimensional program, *FLAC*.* *FLAC3D* extends the analysis capability of *FLAC* into three dimensions, simulating the behavior of three-dimensional structures built of soil, rock or other materials that undergo plastic flow when their yield limits are reached. Materials are represented by polyhedral elements within a three-dimensional grid that is adjusted by the user to fit the shape of the object to be modeled. Each element behaves according to a prescribed linear or nonlinear stress/strain law in response to applied forces or boundary restraints. The material can yield and flow, and the grid can deform (in large-strain mode) and move with the material that is represented. The explicit, Lagrangian calculation scheme and the mixed-discretization zoning technique used in *FLAC3D* ensure that plastic collapse and flow are modeled very accurately. Because no matrices are formed, large three-dimensional calculations can be made without excessive memory requirements. The drawbacks of the explicit formulation (i.e., small timestep limitation and the question of required damping) are overcome by automatic inertia scaling and automatic damping that does not influence the mode of failure. *FLAC3D* offers an ideal analysis tool for solution of three-dimensional problems in geotechnical engineering.

FLAC3D is designed specifically to operate on Microsoft Windows systems, and is currently supported on Windows XP (including XP64) and Windows Vista (including Vista 64). Itanium-based systems are not supported. Calculations on realistically sized three-dimensional models in geo-engineering can be made in a reasonable time period. For example, a model containing 125,000 zones of a Mohr-Coulomb material can be generated within 400 MB RAM. The runtime to perform 5000 calculation steps for a 10,000 zone model of Mohr-Coulomb material is roughly 2 minutes on a 2.7 GHz Intel Core i7 CPU.† The number of calculation steps required to reach a solution state with the explicit-calculation scheme can vary, but a solution typically can be reached within 3000 to 5000 steps for models containing up to 10,000 elements, regardless of material type. (The explicit-solution scheme is explained in Section 1 in **Theory and Background**.) With the advancements in floating-point operation speed, and the ability to install additional RAM at low cost, it should be possible to solve increasingly larger three-dimensional problems with *FLAC3D*.

FLAC3D can be operated from either a command-driven mode or a graphics menu-driven mode. The default command-driven mode is very similar to that used by other Itasca software products. You will find that most of the commands are the same as, or three-dimensional extensions of, those in *FLAC*. A menu-driven, graphical user interface is also available in *FLAC3D* for performing plotting, printing and file access.

---

* Itasca Consulting Group, Inc. ***FLAC (Fast Lagrangian Analysis of Continua.)***, Version 6.0, 2008.

† See Section 5 for a comparison of *FLAC3D* runtimes on various computer systems.

With the graphics facilities built into *FLAC3D*, high-resolution, color-rendered plots are generated quite rapidly. We have developed a graphics screen-plotting facility that allows you to instantly view the model during creation from either command-mode or graphics menu-mode. The model can be translated, rotated and magnified on the screen for better viewing. Color-rendered plots of surfaces showing vectors or contours can be made in 3D, and a two-dimensional plane can be located at any orientation and location in the model for the purpose of viewing vector or contour output on the plane. All output can be directed to a black-and-white or color hardcopy device, or to a file.

You will find that *FLAC3D* offers a facility for problem solving similar to the one in *FLAC*. A comparison of *FLAC3D* to other numerical methods, a description of general features and updates in *FLAC3D* Version 3.1, and a discussion of fields of application are provided in the following sections. If you wish to try *FLAC3D* right away, the program installation instructions and a simple tutorial are provided in Section 2.

1.2 Comparison with Other Methods

How does *FLAC3D* compare to the more common method of using finite elements for numerical modeling? Both methods translate a set of differential equations into matrix equations for each element, relating forces at nodes to displacements at nodes. Although *FLAC3D*'s equations are derived by the finite difference method, the resulting element matrices for an elastic material are identical to those of the finite element method (for constant-strain tetrahedra). However, *FLAC3D* differs in the following respects:

1. The "mixed discretization" scheme (Marti and Cundall 1982) is used for accurate modeling of plastic collapse loads and plastic flow. This scheme is believed to be physically more justifiable than the "reduced integration" scheme commonly used with finite elements.

2. The full dynamic equations of motion are used, even when modeling systems that are essentially static. This enables *FLAC3D* to follow physically unstable processes without numerical distress. The approach to provide a time-static solution is discussed in the definition for "Static Solution" given in Section 2.3.

3. An "explicit" solution scheme is used (in contrast to the more usual implicit methods). Explicit schemes can follow arbitrary nonlinearity in stress/strain laws in almost the same computer time as linear laws, whereas implicit solutions can take significantly longer to solve nonlinear problems. Furthermore, it is not necessary to store any matrices, which means: (a) a large number of elements may be modeled with a modest memory requirement; and (b) a large-strain simulation is hardly more time-consuming than a small-strain run, because there is no stiffness matrix to be updated.

4. *FLAC3D* is robust in the sense that it can handle any constitutive model with no adjustment to the solution algorithm; many finite element codes need different solution techniques for different constitutive models.

These differences are mainly in *FLAC3D*'s favor, but there are two disadvantages:

1. Linear simulations run more slowly with *FLAC3D* than with equivalent finite element programs. *FLAC3D* is most effective when applied to nonlinear or large-strain problems, or to situations in which physical instability may occur.

2. The solution time with *FLAC3D* is determined by the ratio of the longest natural period to the shortest natural period in the system being modeled. This point is discussed in more detail in Section 1 in **Theory and Background**, but certain problems are very inefficient to model (e.g., beams, represented by solid elements rather than structural elements, or problems that contain large disparities in elastic moduli or element sizes).