1 INTRODUCTION

1.1 Overview

The Universal Distinct Element Code (*UDEC*) is a two-dimensional numerical program based on the distinct element method for discontinuum modeling. *UDEC* simulates the response of discontinuous media (such as a jointed rock mass) subjected to either static or dynamic loading. The discontinuous medium is represented as an assemblage of discrete blocks. The discontinuities are treated as boundary conditions between blocks; large displacements along discontinuities and rotations of blocks are allowed. Individual blocks behave as either rigid or deformable material. Deformable blocks are subdivided into a mesh of finite-difference elements, and each element responds according to a prescribed linear or nonlinear stress-strain law. The relative motion of the discontinuities is also governed by linear or nonlinear force-displacement relations for movement in both the normal and shear directions. *UDEC* has several built-in material behavior models, for both the intact blocks and the discontinuities, which permit the simulation of response representative of discontinuous geologic (or similar) materials. *UDEC* is based on a "Lagrangian" calculation scheme that is well-suited to model the large movements and deformations of a blocky system.

UDEC also contains the powerful built-in programming language *FISH* (short for *FLAC*ish; *FISH* was originally developed for our two-dimensional, finite-difference, continuum program: *FLAC*). With *FISH*, you can write your own functions to extend *UDEC*'s usefulness. *FISH* offers a unique capability to *UDEC* users who wish to tailor analyses to suit specific needs.

The formulation and development of the distinct element method embodied in *UDEC* has progressed for nearly 50 years, beginning with the initial presentation by Cundall (1971). In 1985, Dr. Cundall and Itasca staff adapted *UDEC* specifically to perform engineering calculations on an IBM-compatible microcomputer. The software is designed for high-speed computation of models containing several thousand blocks. With the advancements in floating-point operation speed and the ability to install additional RAM at low cost, increasingly larger problems can be solved with *UDEC*. For example, *UDEC* can solve a model containing 3000 deformable blocks with 8 degrees of freedom per block on a personal computer with only 12 MB RAM. The solution speed for a model of this size is roughly 15,000 calculation steps per minute on a 2.9 GHz Pentium i7-870 computer.* The calculation speed is essentially a linear function of the number of blocks in a model, and the number of blocks is a linear function of the available RAM on the computer. (See Table 2.1.)

For typical models, consisting of roughly 1500 rigid blocks (or 500 deformable blocks) or fewer, the explicit solution scheme in *UDEC* requires approximately 2000 to 4000 steps to reach a solved state.** For example, a 500 deformable block model run on the computer described above would require roughly 6 seconds to perform 4000 calculation steps. Consequently, typical engineering

---

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

** This will vary depending on the amount of relative motion that occurs between blocks. The explicit solution scheme is explained in Section 1 in **Theory and Background**.

problems involving several hundred blocks and multiple solution stages can be solved with *UDEC* on a microcomputer in a matter of minutes or a few hours.

A comparison of *UDEC* to other numerical methods, a description of general features and new updates in *UDEC* Version 7.0, and a discussion of fields of application are provided in the following sections. If you wish to try *UDEC* right away, the program installation instructions and a simple tutorial are provided in Section 2.

1.2 Comparison with Other Methods

Some commonly asked questions about *UDEC* follow. Is *UDEC* a *distinct* element or *discrete* element program? What is the difference, and what is *UDEC*'s relation to other programs? We provide a definition here which we hope will clarify these matters.

Many finite element, boundary element and Lagrangian finite difference programs have interface elements or "slide lines" that enable them to model a discontinuous material to some extent. However, their formulation is usually restricted in one or more of the following ways. First, the logic may break down when many intersecting interfaces are used; second, there may not be an automatic scheme for recognizing new contacts; and third, the formulation may be limited to small displacements and/or rotation. Such programs are usually adapted from existing continuum programs.

The name "discrete element method" applies to a computer program only if it

(a) allows finite displacements and rotations of discrete bodies, including complete detachment; and

(b) recognizes new contacts automatically as the calculation progresses.

Without the first attribute, a program cannot reproduce some important mechanisms in a discontinuous medium; without the second, the program is limited to small numbers of bodies for which the interactions are known in advance. The term "distinct element method" was coined by Cundall and Strack (1979) to refer to the particular discrete element scheme that uses deformable contacts and an explicit, time-domain solution of the original equations of motion (not the transformed, modal equations).

There are four main classes of computer programs that conform to the proposed definition of a discrete element method. (The classes and representative programs are discussed in more detail in Section 1 in **Theory and Background**.)

1. Distinct Element Programs – These programs use explicit time-marching to solve the equations of motion directly. Bodies may be rigid or deformable (by subdivision into elements); contacts are deformable. *UDEC* falls into this category.

2. Modal Methods – The method is similar to the distinct element method in the case of rigid bodies but, for deformable bodies, modal superposition is used.

3. Discontinuous Deformation Analysis – Contacts are rigid, and bodies may be rigid or deformable. The condition of no-interpenetration is achieved by an iteration scheme; the body deformability comes from superposition of strain modes.

4. Momentum-Exchange Methods – Both the contacts and the bodies are rigid: momentum is exchanged between two contacting bodies during an instantaneous collision. Frictional sliding can be represented.

There are several published schemes that appear to resemble discrete element methods, but which are different in character or are lacking one or more essential ingredients. For example, many publications are concerned with the stability of one or more rigid bodies, using the *limit equilibrium method* (e.g., Hoek 1973, Warburton 1981, Goodman and Shi 1985, and Lin and Fairhurst 1988). This method computes the static force equilibrium of the bodies, and does not address the changes in force distribution that accompany displacements of the bodies.

1.3 General Features

UDEC is primarily intended for analysis in rock engineering projects, ranging from studies of the progressive failure of rock slopes to evaluations of the influence of rock joints, faults, bedding planes, etc. on underground excavations and rock foundations. *UDEC* is ideally suited to study potential modes of failure directly related to the presence of discontinuous features.

The program can best be used when the geologic structure is fairly well-defined (e.g., from observation or geologic mapping). Both manual and automatic joint generators are built into *UDEC* to create individual and sets of discontinuities which represent (in two dimensions) jointed structure in a rock mass. A wide variety of joint patterns can be generated in the model. A screen-plotting facility allows the user to instantly view the joint pattern. Adjustments can easily be made before the final pattern is selected for analysis.

Different representations of joint material behavior are also available. The basic model is the Coulomb slip criterion, which assigns elastic stiffness, frictional, cohesive and tensile strengths, and dilation characteristics to a joint. A modification to this model is the inclusion of displacement weakening as a result of loss in cohesive and tensile strength at the onset of shear failure. A more complex model, the continuously yielding joint model, is also available and simulates continuous weakening behavior as a function of accumulated plastic shear displacement. As an optional feature, the Barton-Bandis joint model is also available at an additional cost. Joint models and properties can be assigned separately to individual, or sets of, discontinuities in a *UDEC* model. Note that the geometric roughness of a joint is represented via the joint material model, even though the plot of discontinuities shows the joint as a straight-line segment.

Blocks in *UDEC* can be either rigid or deformable. There are seven built-in material models for deformable blocks, ranging from the "null" block material (which represents holes – excavations), to the shear and volumetric yielding models (which include strain-hardening/softening behavior and represent nonlinear, irreversible shear failure and compaction). Thus, blocks can be used to simulate backfill and soil materials as well as intact rock.

The basic formulation for *UDEC* assumes a two-dimensional plane-strain state. This condition is associated with long structures or excavations with constant cross-section acted on by loads in the plane of the cross section. Discontinuities, therefore, are considered as planar features oriented normal to the plane of analysis. In addition, *UDEC* offers a plane-stress option, in which the stresses normal to the cross section are zero. This is encountered, for example, in masonry structures loaded only in the plane of the structure. For plane-strain analysis, blocks may exhibit plastic yield, and failure can occur in the out-of-plane direction if the out-of-plane stress, σ_{zz}, becomes a major or minor principal stress.

The explicit solution algorithm in *UDEC* permits either dynamic or static analysis. For dynamic calculations, user-specified velocity or stress waves can be input directly to the model either as an exterior boundary condition or interior excitation to the model. A library of simple dynamic waveforms is also available for input. *UDEC* contains nonreflecting and free-field boundary conditions for dynamic analysis.

Both stress (force) and fixed displacement (zero velocity) boundary conditions are available for static analysis. Boundary conditions may be different at different locations. In addition, a boundary element model is available to link to the *UDEC* model to simulate the boundary as an infinite elastic body. A half-plane solution is also available to represent the effect of a free surface.

UDEC is able to simulate the flow of fluid through the discontinuities and voids in the model. At present, blocks are impermeable. A fully coupled mechanical-hydraulic analysis is performed, in which fracture conductivity is dependent on mechanical deformation of the joint aperture; conversely, joint water pressures affect the mechanical behavior. Flow is idealized as laminar viscous flow between parallel plates. A visco-plastic flow model is also available to simulate flow of cement grout in the joints.

Structural element logic is implemented to simulate rock reinforcement and surface support in the model. Reinforcement includes point-anchored and fully grouted cables and bolts. Surface support simulates structures such as shotcrete, concrete linings, ribs and other forms of tunnel support, and stabilizing lining for open cuts or natural slopes.

There is also a thermal model available in *UDEC*. This model simulates the transient flux of heat in materials, and the subsequent development of thermally induced stresses. The heat flux is modeled by either isotropic or anisotropic conduction. Heat sources can be added, and can be made to decay exponentially with time.

UDEC contains a powerful built-in programming language, *FISH*, which enables the user to define new variables and functions. *FISH* is a compiler; programs entered via a *UDEC* data file are translated into a list of instructions stored in *UDEC*'s memory space. These are executed whenever a *FISH* function is invoked. *FISH* permits:

- user-prescribed property variations in the grid (e.g., nonlinear increase in modulus with depth);
- plotting and printing of user-defined variables (custom-designed plots);
- implementation of special joint generators;
- servo control of numerical tests;
- specification of unusual boundary conditions; variations in time and space; and
- automation of parameter studies.

An extensive plotting facility is built directly into *UDEC*. This allows the user to generate plots of virtually any problem variable in the *UDEC* model, either on the screen or a hardcopy device. Several variables can be plotted as overlays on a plot of the model or, alternatively, histories of the change in a variable as a function of calculation step can be plotted. The history plots are especially helpful in ascertaining when an equilibrium state or failure state has been reached, and monitoring the change in variables during transient calculations, such as fluid flow in joints. As mentioned previously, plots can be custom-designed via *FISH* to meet the user's need.

1.4 What's New in *UDEC* 7.0

The development that went into *UDEC* 7.0 had two main objectives. The first was to implement a new graphical interface (GUI) for the command line users of *UDEC* . For those familiar with the interface in the Itasca codes PFC, Flac3D, and 3DEC, this interface will be familiar. The second objective was to create a common command syntax that is consistent across all Itasca codes. This resulted in a change to all commands in *UDEC* . There are also many new model building features in UDEC.

1.4.1 New user interface

You now have access to the best of two worlds: the familiar *UDEC* GIIC (Graphical Interface for Itasca Codes) with interactive modeling tools and the new *UDEC* 7.0 GUI (Graphics User Interface), which incorporates the interface used by 3DEC, with advanced graphics and powerful pre- and post-processing tools. Although project files are unique to each of these platforms, *UDEC* 7.0 data and save files can be readily exchanged by each program.

The new *UDEC* 7.0 GUI (shown below) is used in all our three-dimensional programs (FLAC3D, 3DEC, PFC3D). All aspects of the program can be accessed via the main menu (1). The Project Pane lists all the data files (1) and saved files (2) associated with the project. Click on any one to view the data file or restore a previous saved state. Multiple plots can be created, cloned, saved, and viewed via view pane (4). The main tool bar (5) is contextual and will change depending on what pane is active (e.g., text editor pane, view pane, console pane, etc.). There is a new built-in text editor (6) with everything needed to create *UDEC* data files and FISH functions; multiple documents can be associated with a project and navigated by using the tabs (7). Commands may be entered directly via the command line (8), with output displayed in the console pane above. The Control Panel along the far right-hand side changes depending on the active pane. In the image below, plot items (10) are shown since a view pane is active. New plot items may be created, plot ranges can be applied to each one, and they may be copied, pasted, and reordered within a plot view. Each plot item possesses a series of attributes (11) that can be modified. With the mouse cursor over the view pane, model information (object ID - block, zone, structural element, cursor position, value of plot item (label or contour) is displayed (12). Multiple preset GUI layouts are available; custom layouts may be saved.

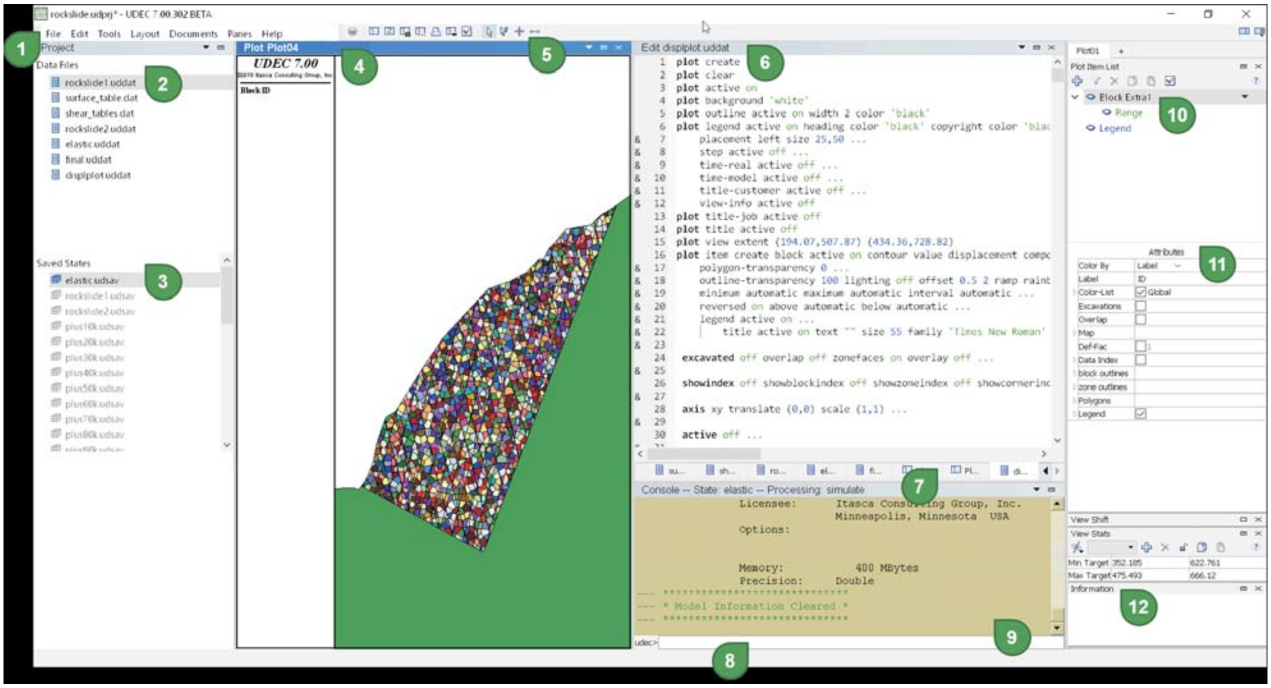

Figure 1.1 New GUI Interface

Features of the *UDEC* 7.0 GUI include: floating panes, including a local control panel for floated panes; expanded plotting of structural element results; measurement tools for distance; cursor pop-up details of plots (position, relevant value measures, object identifiers, etc.); improved movie pre-processing using a dialog to automatically generate movie frames given a list of plots and a list of save files; improved image exporting (PNG, DXF, VRML, PDF, PS, and SVG formats); plots can also be exported as a data file that can be reused by calling into other projects or customized (via commands and FISH); more colors to plot with; built-in file browser shows all files in the project (and other) directories; better control of the view camera, including pre-defined views, saving custom views, and locking views; streamlined plot-item selection and control, including copy-and-paste and plot-item reordering; and pre-defined image sizes have been added to the bitmap output options for movie frame generation.

1.4.2 New Command syntax

All commands are reformulated in *UDEC* 7.0 (GUI and GIIC) to follow a syntactical pattern of NOUN - VERB - OPTION - MODIFIERS - RANGE. Commands are more explicit and more intuitive. They are easier to learn and apply, and are consistent in usage with other Itasca numerical modeling codes.

For example,

UDEC 6.0 Syntax: block 0,0 0,10 10,10 10,0

UDEC 7.0 Syntax: block create polygon 0,0 0,10 10,10 10,0

Commands and keywords can be truncated, so the *UDEC* 7.0 command b cr p 0,0 0,10 10,10 10,0 also works.

In recognition that this major change will require relearning commands and rewriting older data files for compatibility with version 7.0, additional HELP support has been added and a built-in conversion tool is available. The tool automatically converts a specified *UDEC* 6.0 data file (and *UDEC* GIIC project files) to the new *UDEC* 7.0 command syntax. In the event that something is unclear and cannot be automatically converted, it is highlighted for user review. However, the conversion success rate is high and most data files are converted with no flagging whatsoever. When an older data file is updated with the conversion tool, a backup copy of the original data file is automatically retained.

The converter is invoked by loading the file into the built-in file editor and selecting edit->command conversion.

The built in editor will provide prompts for keywords to commands by pressing CTRL-SPACE. Valid FISH commands are highlighted in the editor. Also pressing F1 while typing commands will open a help window for that command.

There is a section of the *UDEC* GUI Help that provides a mapping from old to new *UDEC* commands (see “UDEC 6.0 to 7.0 Commands Map” and “UDEC 6.0 to 7.0 FISH Mapping”).

1.4.3 Added FISH functionality

FISH is a powerful scripting language that can be used to manipulate model components, parameterize models, control model runs, create/calculate new model outputs, monitor results, and post-process model runs. Many new capabilities and tools have been added to *UDEC* to facilitate working with FISH. See Program-Guide->FISH scripting in the *UDEC* GUI Help.

New Data Types In addition to the previous data types (integer, floating-point, string, and pointers), FISH variables now include several new data types: Boolean: Either a value of true or false. Vector: 2D or 3D vector of floating-point types. Array: A collection of FISH variables with specified dimensionality. Matrix: Matrix of numeric values with specified dimensionality. Tensor: A symmetric tensor (e.g., stress and strain). Map: An associative array with string or number key and any FISH variable as value. Structure: A structure may contain multiple FISH variables.

New and expanded data type functions have also been added (e.g., built-in functions to inverse or multiply matrices). FISH Zone Metrics Functions Get one of the metrics of the zone shape (0 - 1, where 1 is ideal) for quality check (ratio of the inscribed radius to the circumscribed radius, minimum ratio of the distance of vertex to the opposing edge divided by the length of the opposing edge length, and ratio of the shortest edge length divided by the longest edge length).

FISH Email Functions (mail.) Compose and send email messages using data files and FISH functions (e.g., send notification emails upon completion of notable tasks, include file attachments such as exported bitmaps to see your results). Email account settings provided with this command may also be specified in the “Email” section of the “Tools/Options” dialog.

Other Enhancements ... FISH variables can now be specified as either local (to a particular FISH function) or global (*UDEC* 6 convention). Inline FISH permits snippets of FISH scripting to be inserted within commands in lieu of FISH variables. FISH functions to create boundary conditions. FISH functions for boundary reaction forces. Working with ground support (structural elements) in *UDEC* is now easier with defined headers to travel the data structure rather than memory offsets.

Know exactly what's happening in your model, even when solving, with the FISH Global Symbols Display Pane.

1.4.4 Built in Editor (GUI)

The Editor pane in *UDEC* provides the ability to edit text-based project resources (data files, FISH files). Though users may choose to work with other text editors for creation/modification of project items with no loss of capability, the *UDEC* editor provides advantages that are not available in outside editors, including: automatic syntax color-coding; debugging and syntax error detection (step-through, check function and variable values) with line number referencing collapsible FISH blocks to simplify reading large, complex data files; columnar selection; undo/redo commands; find/replace dialog; automatically comment (or uncomment) all selected lines (i.e., adds a semicolon to the start of the line, causing it to be ignored upon data file execution); execute only those lines that are currently selected; access to the "Execute/Stop" command, which provides an integrated environment for edit and cycle sequences without having to switch between two programs; and integration with the program documentation via key commands [F1] or [Ctrl+spacebar] to go directly to documentation of the command or FISH function on the current line. Many aspects of the text editor may be customized to suit personal preferences (e.g., font style, font size, syntax colors, etc.) under the Options menu.

1.4.5 Easy Stress Initialization

Easy Model Stress Initiation. Topographic in-situ stress initialization - specify stress gradient - allows you to start out close to equilibrium with the fewest number of blocks or zones or risking artificially activating/shearing more shallow joints by using an initial average in-situ stress value. Automatic stress initialization permits you to define stress at a point directly (e.g., in-situ stress measurement) to define stresses for the entire model (block insitu and block edge commands).

1.4.6 New Block Creation Tools

Built-in automatic block generation commands to create bricks and asymmetric Voronoi blocks.

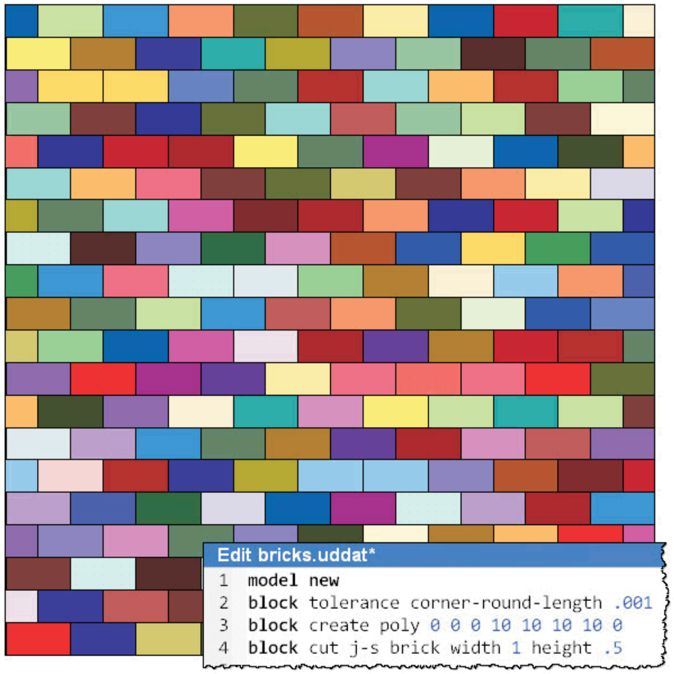

Figure 1.2 Creation of UDEC Brick model

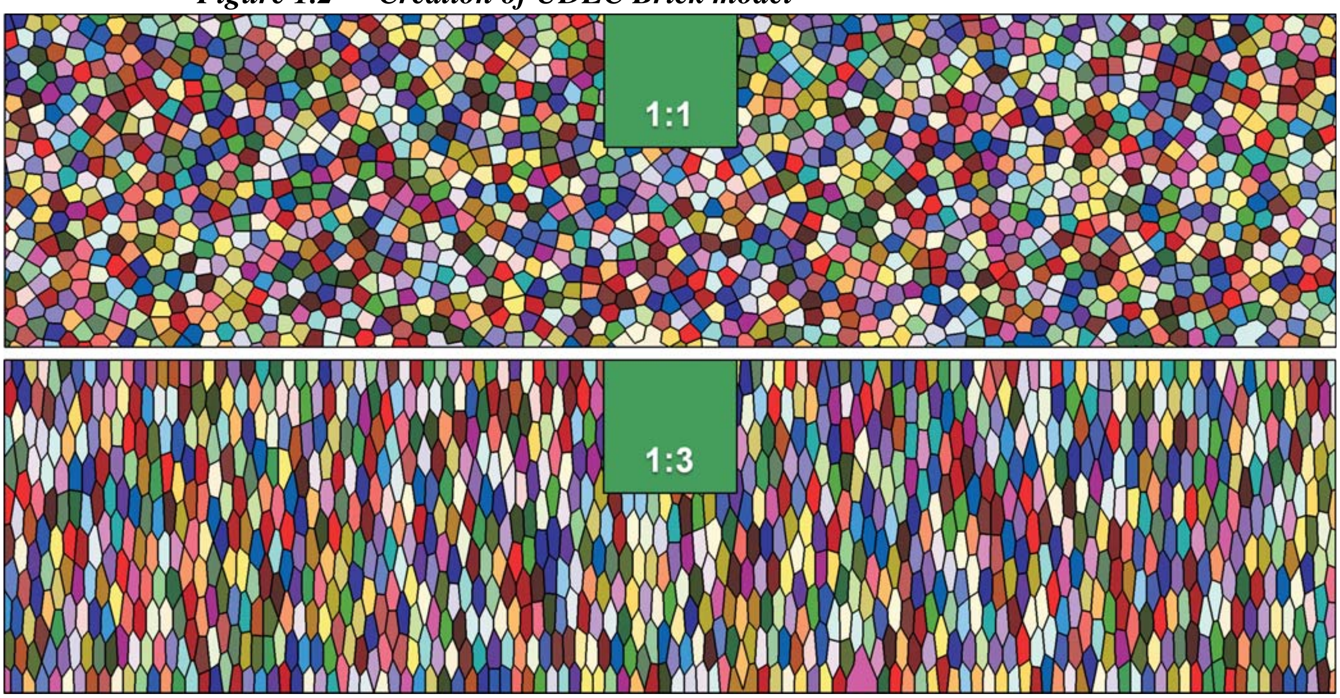

Figure 1.3 Asymmetric Voronoi Block Generation

1.4.7 Import Geometries (such as DXF)

The geometry logic (Help->Program Guide->Common Commands nd FISH->Geometry) allows the user to create, import, and export geometric data. These data can be used to interact with the model in a variety of ways. The data can be used as a filter with the range logic, and as a visualization aid with plotting. FISH extra variables and group names may be assigned to geometric data. FISH may be used to create and manipulate the data. The geometric data are organized into geometric sets, which are named collections of nodes, edges, and polygons. The geometry import command is used to import geometric data from a file that can be of type STL, DXF, or the Itasca geometry format.

Using the Geometry logic, you can now import and use DXF files for *UDEC* model construction with automatic conversion of geometry into crack commands. DXFs are required to be watertight and of good quality. Some interactive DXF clean-up tools are available in the GIIC.

1.4.8 Rock Support Additional Features

In addition to improved visualization (GUI) for structural elements, a rock support database, including fully grouted bolts, end-anchor bolts, Swellex bolts, and split-set bolts (GIIC) is included. More examples for replicating manufacturer pull-test results have also been included (GUI/GIIC) to aid with model calibration.

1.4.9 Groups, Extra Variables

Up to 128 extra variables are now available for all objects (blocks, zones, grid points, structural elements, contacts, etc.). See Help->Program Guide->Common Commands and FISH->Extra. The group logic is the tool used to categorize model objects. Up to 128 different slot names can now be assigned to each group. A slot can be thought of as an independent container of groups. See Help->Program Guide->Common Commands and FISH->Group.

1.4.10 3D joint specification

The angle keyword is used to specify orientation of jointing. With *UDEC* 7.0, you can also specify 3D joint orientations (dip and dip-direction) and the 2D model plane, with *UDEC* 7.0 automatically calculating the joint-plane intersection traces as visualized in the following 3DEC model. This permits you to rapidly generate a 2D model or a series of models using real-word data without having to do any math yourself.

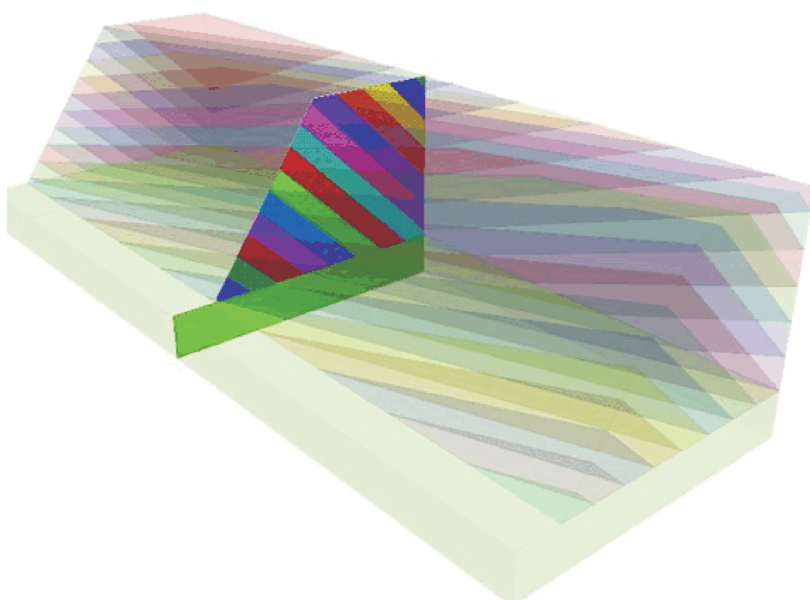

Figure 1.4 Creation of UDEC model using 3D joint Definition

1.4.11 Discrete Fracture Network (DFD)

UDEC 7.0 also includes DFN logic with a built-in DFN generator based on a statistical description. Fractures are generated until a specified stopping criterion is met. These conditions include: a target fracture number; a target P10 (defined from borehole data as fracture frequency); a target density P21 (from trace mapping upon surfaces such as benches or tunnel wall); a target percolation volume; DFN at the connectivity threshold; and user-defined criteria (fish-stop). The logic is similar to that used in 3DEC and also permits the simulation of joints embedded within blocks, whereas joint commands would truncate (remove) joint segments that fall within blocks. This is achieved by cutting a construction joint in the same position/orientation as the fracture until it intersects the model boundary or another joint. Different properties can then be assigned to the contacts falling on fractures (e.g., zero cohesion) and “solid” rock (e.g., contacts outside of fractures are “glued”). New plot items for working with DFNs, including stereonet and rosette charts, have also been added. The DFN logic is fully supported by both commands and FISH functions. See Help->Program Guide->Common Commands and FISH->Discrete Fracture Network (DFN).

1.4.12 Factor of Safety Contours

UDEC uses a factor of safety calculation for stability analyses based upon the “strength reduction metho”. A “factor of safet” index can be defined for any relevant problem parameter by taking the ratio of the calculated parameter value under given conditions to the critical value of the parameter, at which the onset of an unacceptable outcome manifests itself. Typically, application of the strength reduction method produces one single factor of safety per simulation, corresponding to one global minimum stability state. However, the ability to calculate multiple minimum states may be of interest, for example, to generate a safety map for a complex slope profile such as a benched cut or a slope with a berm (see following figure). The explicit dynamic solution method employed in *UDEC* allows multiple local stability surfaces to be identified in one simulation.

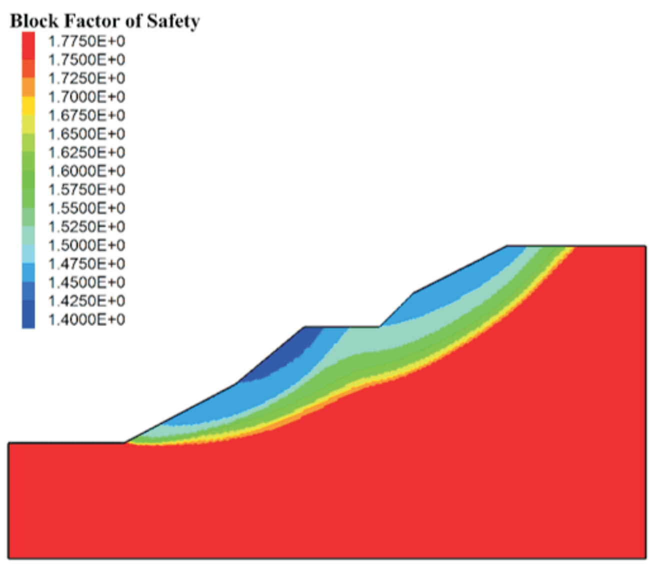

Figure 1.5 Factor of Safety Contours

With factor of safety contouring, material strengths are reduced in increments by a strength reduction factor. Unstable states for the model are identified at the global minimum state and then beyond that state. Unstable states of the model are identified at each stage as an assembly of gridpoints with velocities above a specified average value. The current strength reduction factor is assigned to unstable gridpoints for later contouring. If the strength is reduced in small intervals, progressively more regions of gridpoints can be identified as unstable. By monitoring the velocities, it is possible to delineate the regions of unstable gridpoints by different strength factors and produce a plot of factor of safety contours.

1.4.13 Graphical Information Retrieval

When a View pane (plot) is active, the toolbar includes a group of buttons to set the current mouse mode to one of four states: Select, Query, Center, and Distance. When the Query mode is selected, as highlighted in the image below, you can use the mouse cursor to inspect your model. Clicking on a location in the model will provide a list of one or more objects that can be queried (e.g., block, zone, gridpoint). Selecting an object from the menu will open a dialog that contains model information pertinent to it as shown; data can be copied to the clipboard or logged.

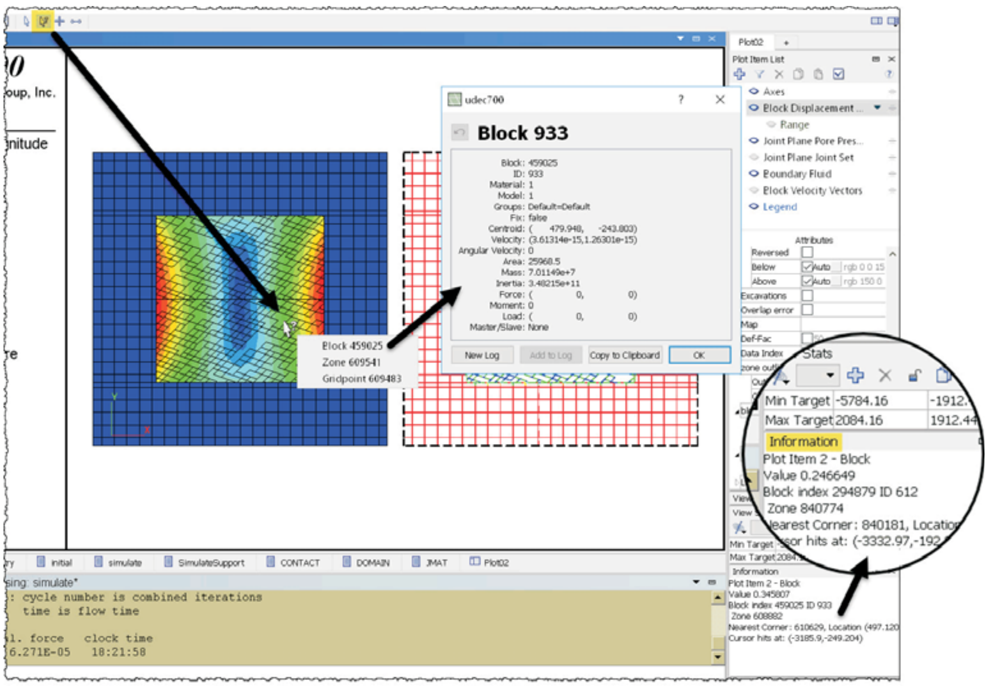

Figure 1.6 Graphical Information Retrieval

1.4.14 Contact Failure States

Added failure states for all contact (joint) constitutive models (e.g., failed tension now, failed shear past). Added command to display contact failure state and contact model. Quickly export chart and table data to CSV files for use with spreadsheet software.

1.4.15 Added future Save File Compatiblity

Updated save/restore system. Enhancements will allow for backward compatibility of SAV files for future versions of *UDEC* . (GUI/GIIC)

1.4.16 GUI project files

A project in *UDEC* (GUI) may now be created. The use of project file in the *UDEC* GUI is highly recomended. The file contains all plots and links to all input and save files. The project file and all data and save files can be bundles for achiving or sharing. See Help->Program Guide->Program Mechanics->Projects.

1.4.17 Technical Support Dialog

New technical support dialog added to the HELP menu that lets users submit technical support requests and automatically includes license and system information that helps Itasca help you (GUI).

1.4.18 Dynamic Input Wizzard

The Dynamic Input Wizard tool facilitates the preprocessing of input signals necessary to correct signal input for use in seismic and dynamic analyzes. The wizard includes the following functionality: Import ground motion data (velocity or acceleration) from a *UDEC* history, *UDEC* table, or PEER file, and specify data units. Filter ground motion data to remove high-frequency components with a specified cut-off frequency. Add a baseline correction for velocity or displacement drift by removing the mean acceleration, the running average displacement, a polynomial fit displacement, or a low-frequency sine function from the acceleration or velocity. Export processed data (acceleration, velocity, displacement, amplitude spectrum, response spectrum, or Arias intensity) as a *UDEC* history or *UDEC* table file. See the dynamic section in the "Special Features Manual".

1.4.19 Updated save/restore system

Enhancements will allow for backward compatibility of SAV files for future versions of *UDEC* .

1.5 Fields of Application

UDEC was originally developed to perform stability analysis of jointed rock slopes. The discontinuum formulation for rigid blocks and the explicit time-marching solution of the full equations of motion (including inertial terms) facilitate the analysis of progressive, large-scale movements of slopes in blocky rock.

UDEC has been applied most often in studies related to mining engineering. Both static and dynamic analyses for deep underground mined openings have been performed. Fault-slip induced failure around excavations is one example of analyses conducted with *UDEC*. Blasting effects have been studied by applying dynamic stress or velocity waves at model boundaries. Research in the area of fault-slip induced seismicity has also been conducted by use of the continuously yielding joint model. Structural elements have been employed to simulate various rock reinforcement systems, such as grouted rockbolting and shotcrete.

UDEC has also been applied in the fields of underground construction and deep underground storage of high-level radioactive waste. Through the use of the thermal model, *UDEC* has been used to simulate effects of thermal loading in connection with buried nuclear waste.

UDEC has been used to a limited extent as a computational design tool. However, the program is better-suited to investigate potential failure mechanisms associated with the response of a jointed rock mass. The nature of a jointed rock mass is that it is a “data-limited” system (i.e., the internal structure and stress state are, in large part, unknown and unknowable). Thus, it is impossible, in principle, to make a complete model of a rock mass system. Also, since *UDEC* is a two-dimensional program, the three-dimensional geometry of a joint structure cannot be represented except for special orientations. Nevertheless, an understanding of the response of underground openings in jointed rock can be achieved at a phenomenological level using *UDEC*. This methodology seeks to improve the engineering understanding of the relative impact of various phenomena on the rock mechanics design. In this way, the engineer can anticipate potential problem areas by identifying mechanisms that may lead to unacceptable states of deformation/loading (or failure) of the underground opening. The paper by Starfield and Cundall (1988) is recommended as a guide for using *UDEC* in rock engineering projects.

UDEC has potential for application in other fields of engineering, as discussed below.

The fluid flow model in *UDEC* has been used for studies of fluid penetration from unlined pressure tunnels, storage of natural gas in rock caverns, and flow through jointed rock foundations beneath gravity dams. The visco-plastic formulation of the flow model has been employed to simulate flow of cement grout. Examples of these applications are provided in the **Verification Problems**.

UDEC also has the potential for application in studies related to earthquake engineering. For example, the program may be used to provide explanations of phenomena related to fault movement.

Another area of application is the study of the behavior of reinforced concrete. Although *UDEC* does not include a model to simulate dynamic fracture growth through deformable blocks, progressive failure associated with crack propagation and spalling can be simulated by the breaking of preexisting bonds between blocks which remain intact. A special joint generator which creates Voronoi polygons within the *UDEC* model is employed (Lorig and Cundall 1987).

It is important to note that *UDEC* is not a suitable program for particle flow studies or dynamic analysis of cratering phenomena in which the interaction of many microparticles is important. For these studies, programs such as the Itasca code *PFC*^{2D} are recommended. The new cell space detection logic in *UDEC* does allow for the analysis of flying blocks, but is not efficient for problems with thousands of tiny particles. Finally, in order to evaluate the importance of three-dimensional geometry on response of a system, a three-dimensional numerical program, such as the Itasca code *3DEC* or *PFC*^{3D}, is required.

1.6 Guide to the Manual

The *UDEC* Version 7.0 manual consists of seven volumes. These are provided with *UDEC* as PDF files. The Command Reference and Fish Reference volumes have been moved to the Help in *UDEC* .

User's Guide

Section 1 Introduction

This section introduces you to *UDEC* and its capabilities and features.

Section 2 Getting Started

If you are just beginning to use *UDEC*, or are only an occasional user, we recommend that you read Section 2. This section provides instructions on installation and operation of the program, as well as recommended procedures for running simple *UDEC* analyses.

Section 3 Problem Solving with *UDEC*

Section 3 is a guide to practical problem solving. Turn to this section once you are familiar with the program operation. Each step in a *UDEC* analysis is discussed in detail, and advice is given on the most effective procedures to follow when creating, solving and interpreting a *UDEC* model simulation.

Section 4 *FISH* Beginner's Guide

Section 4 provides the new user with an introduction to the *FISH* programming language in *UDEC*. This includes a tutorial on the use of the *FISH* language. *FISH* is described in detail in Section 1 in the ***FISH* volume**.

Section 5 Miscellaneous

Various information is contained in this section, including the *UDEC* runtime benchmark and procedures for reporting errors and requesting technical support.

Theory and Background

Section 1 Background – The 2D Distinct Element Method

The theoretical formulation for *UDEC* is described in detail in Section 1 in **Theory and Background**.

Section 2 Factor-of-Safety Calculation

The steps and options for calculating factors of safety are described.

Section 3 Energy Calculation

The stored and dissipated energy components that can be monitored in *UDEC* are described. An example application is given.

Constitutive Models

Section 1 Block Constitutive Models

The theoretical formulation and implementation of the various block constitutive models are described.

Section 2 Continuously Yielding Joint Model

The formulation of the continuously yielding joint model is described, and example applications are provided.

Section 3 Barton-Bandis Joint Model

The implementation of the Barton-Bandis joint model in *UDEC* is described. The input commands are given for this option, along with examples to illustrate the application of the model.

Section 4 Writing New Constitutive Models

This section describes the process that allows users to write their own zone and joint models, which are compiled and included as DLLs.

Creep Material Models

Section 1 Creep Material Models

The different creep models (available as an option in *UDEC*) are described, and verification problems are provided.

Special Features – Structures/Fluid Flow/Thermal/Dynamics

Section 1 Structural Elements

This section describes the various structural element models available in *UDEC*.

Section 2 Fluid Flow in Joints

The formulation for the joint fluid-flow model is described, and the various ways to model transient and steady-state flow in joints are illustrated.

Section 3 Thermal Analysis

The thermal analysis facility in *UDEC* is described, and several verification problems illustrating its application (both with and without interaction with mechanical stress) are presented.

Section 4 Dynamic Analysis

The dynamic analysis facility is described, and considerations for running a dynamic model are provided. Several verification examples are also included in this section.

Verification Problems

This section contains a collection of *UDEC* verification problems. These are tests in which a *UDEC* solution is compared directly to an analytical (i.e., closed-form) solution.

Example Applications

This section contains example applications of *UDEC* that demonstrate the various classes of problems to which *UDEC* may be applied.

1.7 Itasca Consulting Group Inc.

Itasca Consulting Group Inc. is more than a developer and distributor of engineering software. Itasca is a consulting and research firm consisting of a specialized team of civil, geotechnical and mining engineers with an established record in solving problems in many areas:

Civil Engineering

Mining Engineering and Energy Resource Recovery

Nuclear Waste Isolation and Underground Space

Defense Research

Software Engineering

Seismic Engineering

Groundwater Analysis and Dewatering

Petroleum Engineering

Itasca was established in 1981 to provide advanced rock mechanics services to the mining industry. Today, Itasca is a multidisciplinary geotechnical firm with over 100 professionals in offices worldwide. The corporate headquarters for Itasca is located in Minneapolis, Minnesota. Worldwide offices of Itasca are: Itasca Denver Inc. (Denver, Colorado); Itasca Consultants AB (Luleå, Sweden); Itasca Consultants S.A.S. (Ecully, France); Itasca Consultants GmbH (Gelsenkirchen, Germany); Itasca Consultores S.L. (Llanera, Spain); Itasca S.A. (Santiago, Chile); Itasca Consulting Canada Inc. (Sudbury, Canada); HydroChina – Itasca R & D Center (Hangzhou, China); Itasca Australia Pty. Ltd. (Melbourne, Australia); Itasca Consulting Ltd. (Shrewsbury, United Kingdom); Itasca India Consulting Pvt. Ltd. (Nagpur, India); Itasca Perú (Lima, Peru); and IDI-South Africa (Rivonia, South Africa).

Itasca's staff members are internationally recognized for their accomplishments in geological, mining, petroleum, seismology and civil engineering projects. Itasca staff consists of geological, mining, hydrological, petroleum and civil engineers who provide a range of comprehensive services such as (1) computational analysis in support of geo-engineering designs, (2) design and performance of field experiments and demonstrations, (3) laboratory characterization of rock properties, (4) data acquisition, analysis and system identification, (5) groundwater modeling, and (6) short courses and instruction in the geomechanics application of computational methods. If you should need assistance in any of these areas, we would be glad to offer our services.

1.8 User Support

We believe that the support Itasca provides to code users is a major reason for the popularity of our software. We encourage you to contact us when you have a modeling question. We will provide a timely response via telephone, email or fax. General assistance in the installation of *UDEC* on your computer, plus answers to questions concerning capabilities of the various features of the code, are provided free of charge. Technical assistance for specific user-defined problems can be purchased on an as-needed basis.

If you have a question, or desire technical support, please contact us:

Itasca Consulting Group Inc.
Barrel House
111 Third Avenue South, Suite 450
Minneapolis, Minnesota 55401 USA

Phone:	(+1) 612-371-4711
Fax:	(+1) 612·371·4717
Email:	software@itascacg.com
Web:	www.itascacg.com

We also have a worldwide network of code agents who provide local technical support. Details may be obtained from Itasca.

1.9 References

Cundall, P. A. "A Computer Model for Simulating Progressive Large-Scale Movements in Blocky Rock Systems," in ***Proceedings of the Symposium of the International Society for Rock Mechanics (Nancy, France, 1971)***, Vol. 1, Paper No. II-8 (1971).

Cundall, P. A., and O. D. L. Strack. "A Discrete Numerical Model for Granular Assemblies," *Geotechnique*, ***29***, 47-65 (1979).

Goodman, R. E., and G.-H. Shi. ***Block Theory and Its Application to Rock Engineering***. New Jersey: Prentice Hall (1985).

Hoek, E. "Methods for the Rapid Assessment of the Stability of Three-Dimensional Rock Slopes," *Quarterly J. Eng. Geol.*, ***6***(3) (1973).

Lin, D., and C. Fairhurst. "Static Analysis of the Stability of Three-Dimensional Blocky Systems around Excavations in Rock," *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.*, ***25***(3), 138-147 (1988).

Lorig, L. J., and P. A. Cundall. "Modeling of Reinforced Concrete Using the Distinct Element Method," in ***Fracture of Concrete and Rock***, pp. 459-471. S. P. Shah and S. E. Swartz, eds. Bethel, Conn.: SEM (1987).

Starfield, A. M., and P. A. Cundall. "Towards a Methodology for Rock Mechanics Modelling," *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.*, ***25***, 99-106 (1988).

Warburton, P. M. "Vector Stability Analysis of an Arbitrary Polyhedral Rock Block with Any Number of Free Faces," *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.*, ***18***, 415-427 (1981).