WinGridder

An interactive grid generator for TOUGH

Version 1.0

(User's Manual)

Lehua Pan, Jennifer Hinds, Charles Haukwa, Yu-Shu Wu, and Gudmundur Bodvarsson

Earth Sciences Division

Ernest Orlando Lawrence Berkeley National Laboratory

July 2001

Table Contents

Executive Summary

WinGridder V1.0 is designed to generate a 3-D grid (single or dual continuum) that conserves the physical relationships between objects to be simulated with a finite number of cells. It can save mesh files that can be used as input files of TOUGH family codes. It can also output additional grid information for various purposes. It has user-friendly graphical user interfaces. With its interactive editing functions, a user can easily check and modify a layout of columns.

This manual describes the approaches, the architecture, and the functionality of WinGridder, complemented with several examples. Relying on this manual, users with adequate knowledge of numerical gridding and numerical modeling (especially of integrated finite difference method) should be able to generate numerical grids for the TOUGH family of codes. Although the developer does not offer any service, any questions about usage of the software or suggestions for improving the software are welcome. The users are encouraged to contact Dr. Lehua Pan at

Email: LPAN@LBL.GOV
Phone: (510) 495-2360.
Mail: MS 90-1116
Earth Science Division
Lawrence Berkeley National Laboratory
1 Cyclotron Road
Berkeley, CA 94720

1. Introduction

1.1 Background

WinGridder is a Windows-based software package for designing, generating, and visualizing at various spatial scales numerical grids used in reservoir simulations and groundwater modeling studies. Development of this software was motivated by the requirements of the TOUGH (Transport of Unsaturated Groundwater and Heat) family of codes (Pruess 1987, 1991) for simulating subsurface processes related to high-level nuclear waste isolation in partially saturated geological media. Although the TOUGH family of codes has great flexibility in handling the variety of grid information required to describe complex objects, designing and generating a suitable irregular grid can be a tedious and error-prone process, even with the help of existing grid generating programs. This is especially true when the number of cells and connections is very large. The processes of inspecting the quality of the grid or extracting sub-grids or other specific grid information are also complex. The mesh maker embedded within TOUGH2 generates only uniform numerical grids and handles only one set of uniform fracture and matrix properties throughout the model domain. This is not suitable for grid generation in complex flow and transport simulations (such as those of Yucca Mountain, which have heterogeneity in both fracture and matrix media). As a result, the software program “Amesh” (Haukwa 2000) was developed to generate irregular, effective-continuum (ECM) grids.

However, there are three major shortcomings with Amesh. First, Amesh cannot generate grids with inclined faults. Second, it does not provide functions for assigning nodes to represent important geological and engineering structures, such as adding embedded refinements to the grid to capture (for example) waste-emplacement-drift spacing within the potential repository at Yucca Mountain. Third, it has no capabilities for visualizing, editing, and checking the generated grids. These limitations precipitated the development of WinGridder, a user-friendly, efficient, and effective grid-generating software package for use with the TOUGH family of codes.

1.2 Objectives

The objectives of WinGridder are twofold:

(1) To provide an efficient, user-friendly, integrated gridding package with graphical interfaces and visual inspection capabilities

(2) To generate a grid that conserves the physical relationships between objects to be simulated and that is capable of having a local refinements and dual continua, with options for representing fractured media.

1.3 How to Use This Manual

It is recommended that the user first read Sections 2 and 3, entitled “General Approaches” and “Special Features”, respectively, to become familiar with the basic concepts and terms used in WinGridder. “General Approaches” summarizes the object structures used to describe a grid in WinGridder and provides the general steps the user should adopt to generate a grid with WinGridder. “Special Features” discusses how WinGridder generates a grid that conserves important geological and engineering features. After acquiring this basic understanding of the software, the user should proceed to Section 4 of the manual, “WinGridder Architecture”. This section gives the user an overview of WinGridder, which contains important information about file structure and formats. The user must be familiar with the input files and have them formatted prior to using WinGridder to generate a grid. It is also necessary to know the formats of the various output files to extract the desired information. Section 5, “Functionality”, is the major section of this manual, providing detailed instructions about how to use WinGridder. Many of the gridding functions are straight forward, and no additional illustration is provided. Other functions are more complex, for which examples have been included. Section 6, “Examples”, documents applications of various software functions for grid generation. Some examples are from unsaturated zone (UZ) modeling studies of Yucca Mountain and may be used to meet specific gridding needs.

2. General Approaches

WinGridder is used to design and generate grids for numerical simulators that are based on the "integral finite difference" or the "control volume" numerical scheme (e.g., TOUGH2, Pruess 1991). Such grids discretize the simulation domain into a collection of cells connected by the interfaces between them (Figure 2-1). A cell is a control volume over which the governing equations will be integrated, and it is also the smallest unit that holds all primary variables (e.g., saturation, pressure) and conservation laws in a grid. Each cell has properties that represent the geological and geometric features of the subsurface domain that it represents (Figure 2-2a). As an object in the grid, each cell holds information about its neighboring cells as well as the corresponding connections. Furthermore, a cell holds such information as the geological layer and column to which the cell belongs, the kind of cell it is, and its rock properties. The interfaces describe the geometric relationships between cells. Each interface connects two adjacent cells and has its own geometric properties that describe the features of the interface and its relationship to the adjacent cells (Figure 2-2b).

A secondary structure witch consists of columns (Figure 2-2c) and segments (Figure 2-2d) is used in WinGridder to organize grid information (Figure 2-1). This secondary structure is a 2-D grid of the model domain (in map view) that provides the basis for projecting the 3-D grid. The segments divide the domain into columns, which may be either vertical or inclined. Each column consists of a series of cells and corresponding vertical connections, while each segment consists of lateral interfaces between two neighboring columns. If local refinement is involved, the lateral connections between those refined cells within a column still belong to the column in the secondary structure. Each column has a representative node in the 2-D, map-view grid, indicating the column's position in the x-y plane. Each column holds the names of the top cell face and bottom cell face to represent itself in a 3-D grid. A similar approach is used for segments. Note that the word "node," as used in this manual during the designing stage of grid generation, is synonymous with "column." Additionally, a column is a polygon in the x-y plane whose boundaries consist of segments that are defined prior to 3-D grid generation.

Grid development starts with the assignment of nodes in map view for each object [e.g., the domain (base nodes), faults, the repository] with specified orientation and density. Based on the location of these nodes, a primary 2-D grid is generated with Voronoi tessellation techniques embedded in Amesh (Haukwa 1998). The 2-D grid is then improved systematically and interactively by deleting physically incorrect or unnecessary connections. A few iterations of these steps, including adding and deleting certain nodes, are sometimes necessary to create a final 2-D grid (a column scheme) that serves as the basis for generating the third dimension of the grid. All 3-D cells and vertical connections between adjacent cells are generated column by column to ensure that each vertical connection connects only adjacent cells and that each cell has at least one vertical connection. Lateral connections are then generated segment by segment in map view, with each segment joining two neighboring columns. This ensures that only cells in two adjacent columns have lateral connections and that no connections between two adjacent columns are missing.

In summary, the first step in building a 3-D grid is to generate a 2-D grid that divides the model domain into connected columns. The second step is to divide each column into a series of connected cells. The third step is to define the interfaces (lateral connections) between adjacent columns on a cell-by-cell basis.

3. Special Features

3.1. Conservation of Stratigraphic Features

Large-scale modeling often involves capturing a series, or stack, of hydrogeologic units (layers). The geometry of each unit is represented by a set of data (either thickness or elevation) on a regular or irregular grid spacing. Using these data, WinGridder generates a numerical grid based on the elevations of three major horizons: (1) a top boundary (e.g., the topographic or bedrock surface), (2) a structural reference horizon, which identifies faults and their associated offsets, and (3) a bottom boundary (e.g., the water table). The reference horizon is the surface from which elevations of all hydrogeologic-unit interfaces are calculated by stacking layer thicknesses above or below it based on their stratigraphic position. All offsets resulting from faulting are described by the reference horizon data. Any portions of hydrogeologic units lying above the top boundary or below the bottom boundary after stacking are removed (clipped).

For a given column, 3-D cells are built for each hydrogeologic unit, first above the reference surface until reaching the top boundary, and then below the reference surface down to the bottom boundary. The interfaces of the generated cells are located exactly at the interfaces of the corresponding hydrogeologic layers at the nearest x, y coordinate for which there is input data. Consequently, it is desirable to have input data on a regular, reasonably fine grid spacing, particularly if the layer geometries are complex or steeply dipping. WinGridder does not interpolate layer thickness or elevation from an input data file; rather, it selects the closest data point and uses its value. This is done to avoid smoothing the offset of geological layers resulting from faulting. Vertical connections are generated after each cell is built, with the reference horizon defining the first vertical connections within a given column. A dummy cell is added to the top and the bottom of each column to allow the user to specify boundary conditions.

If the thickness of a layer is larger than the user-specified maximum thickness, the layer is divided into two or more sublayers, and multiple cells are generated to represent them. Therefore, no cell will have a thickness greater than the specified maximum thickness. Conversely, if the thickness of a layer is less than the user-specified minimum thickness, the layer is considered absent, and no cell will be generated for the layer at this location. However, layer thicknesses (if any) below this threshold will be added to the underlying layer to conserve the total thickness of the stack.

When building lateral connections, two adjacent columns are searched top-to-bottom. If cells in the adjacent columns belong to the same layer, a lateral connection is built for them. The height of the interface is determined by the minimum height of two connected cells, while the length of the connection is the same as the length of the corresponding segment. In some cases, a cell may not have a counterpart in an adjacent column because the layer is pinched-out. In this case, the cell that represents the last occurrence of a particular layer is laterally connected to the cell of the layer above the missing layer in the adjacent column. The height of the interface at the pinch-out margin is reduced to 10% of the user-specified minimum thickness.

Material properties are assigned to cells depending on the hydrogeologic layer to which the cell corresponds. For layers with multiple properties (e.g., the vitric and zeolitic zones within the Calico Hills Formation of Yucca

Mountain), polygons defining the aerial extent of these zones are created by a series of vertex coordinates. Assignment of material properties (e.g., vitric and zeolitic) to model grid cells is then confined to the appropriate polygon by checking if the representative nodes are located inside the polygon.

3.2. Representation of Inclined Faults

A fault can be a surface with arbitrary shape in a 3-D domain and is often defined by a set of (x, y, z) data in geologic models. In grids generated with WinGridder, fault surfaces are represented by a series of connected columns of cells. Faults can be represented in the grid as either vertical or nonvertical features. For vertical faults, only the x, y coordinates of each fault at some arbitrary elevation are needed to define the location of fault columns in a grid generated with WinGridder.

Representation of nonvertical faults is more complex. For convenience in numerical gridding, contour lines (traces) of each fault at three user-specified elevations are used to define its 3-D orientation. Usually, the highest elevation is the projected elevation of the fault just above the top boundary of the modeled domain, while the lowest elevation is taken at some value just below the bottom boundary. The middle elevation is taken at some arbitrary horizon, preferably near the middle (vertically) of the model domain. The gridding process then interpolates the location of each nonvertical fault using data points at these three elevations. With this approach, the dip of a fault within a given fault column is uniform in the upper interval between the highest and middle elevations, and is again uniform in the lower interval between the middle and lowest elevations. This allows the dip in the upper interval to be different from the dip in the lower interval (which may occur if the fault surface is curved, rather than planar). Furthermore, dip angles within the same interval can be different in different columns (i.e., laterally along a fault). Thus, even a fault with variable dip along its trace can be represented with this method. WinGridder allows the user to define the width of each fault zone within the numerical grid.

Conceptually, three important features of a fault are conserved in the numerical gridding formulation of WinGridder. First, a fault is a separator that causes discontinuity of geological layers and may serve as a structural barrier to lateral flow. Second, a fault zone is continuous and may serve as a fast path for flow depending on its hydraulic properties. Third, a fault may or may not be vertical, and its angle of inclination may vary spatially. To implement these features in a grid, three parallel rows of fault-related columns are built for each fault. Each section of a fault in map view consists of three connected columns, with the fault column located in the middle (Figure 3.2-1). Each fault column is connected to two side columns and two neighboring fault columns only. Adjacent fault columns belonging to the same fault will always be connected to each other. Two columns on opposite sides of a fault are always separated by a fault column. Note that for a section of a fault, the incline of the fault becomes a two-dimensional issue on the cross-sectional view. Representation of fault intersections cannot be automatically implemented using WinGridder because each fault column is laterally bound by fault-related side columns. For simplicity, if two faults intersect, the involved fault column will use the orientation of the major fault. If fault intersection is an important feature to capture, the user can manually edit the final mesh file by assigning proper rock properties to the related cells to make such an intersection.

The three fault-related columns (the fault column and its two side columns) are processed together to generate 3-D cells representing the fault and layer offset. From the top boundary to the bottom boundary, the x-y location of fault cells shifts according to the elevation and dip of the fault. Similarly, the volumes and the center (nodal point) location of the corresponding side cells are adjusted accordingly. As a result, the inclination of the fault is described by a series of connected fault cells whose x-y locations vary with elevation. The fault-related cells are vertically connected if they belong to the same column, regardless of the fault angle. Columns of side cells are connected in a similar fashion regardless of the horizontal shifting of position and change in volume. To look at it from another perspective, each set of three fault-related columns (i.e., the fault column plus its two side columns) can be viewed collectively as one vertical column that is subdivided into three nonvertical columns to capture the angle of inclination along a fault.

This method of representing the three-dimensionality of faults requires that all fault grid cells have the same elevation and thickness as their neighboring side cells to facilitate vertical displacement of geologic layers. Because a model domain may be comprised of variably thick hydrogeologic layers, simply reassigning material properties from one row of grid cells to another to establish offset along faults is insufficient for representing the true layer configurations. This approach may remove certain layers from columns adjacent to fault columns and may misrepresent layer thicknesses. To avoid such error, additional resolution is added to fault-related cells based on the location of stratigraphic boundaries on both sides of the fault. Therefore, vertical grid discretization in each set of three fault-related columns is identical, and often much greater than nonfault-related columns. Interfaces between stratigraphic units in both side columns correspond to the interfaces between grid cells (Figure 3.2-1). The layer identification and rock properties of fault cells are then assigned according to the hydrostratigraphic position.

The assignment of lateral connections that involve fault-related cells is different from the way lateral connections are assigned to normal (nonfault-related) cells. Fault-related lateral connections are of two types, fault-fault cell connections and fault-side cell connections. In these two cases, lateral connections occur between cells that share the same interface. The interface area is the contact area between the two cells.

3.3. Representation of the Repository and Local Refinement

The local refinement feature of WinGridder was developed specifically for underground waste repository representation in numerical modeling studies of Yucca Mountain. As such, this feature is discussed in the context of its intended purpose. The repository is defined as a single, 3-D body that is subdivided into blocks to represent individual waste emplacement drifts. It can, however, be used to represent any engineering object that occupies 3-D space and requires special gridding treatment (i.e., local refinement).

The repository area is discretized into a regular mesh of repository columns, except along faults and those cells corresponding to well locations. All repository columns are aligned along the direction of the emplacement drifts, and each column has four sides to facilitate the numerical representation of a drift with a series of connected 3-D cells. For a mountain-scale model grid, the radius of a drift and the spacing between drifts are too small to be represented explicitly, considering the size and number of columns needed and the limitation of computer resources. However, local refinement can be restricted to within the repository horizon. This allows the representation of individual drifts without creating a large number of cells.

Building 3-D cells for a column within the repository boundary is similar to building 3-D cells for any other column outside the repository boundary, except at the elevation of the repository horizon itself. Within the repository horizon, the interfaces between the cells no longer conserve the interfaces between hydrogeologic layers. For each repository column, the elevation of the repository horizon is calculated, and the user specifies a repository-horizon thickness. This thickness is then divided vertically into five equal sections. Without local refinement, each section becomes a 3-D cell with a thickness equal to one-fifth of the repository-horizon thickness. When local refinement is engaged, the second and fourth sections are split laterally into four cells, while the third section (the drift horizon) is split laterally into sixteen cells (Figure 3.3-1). Each cell within the second section connects vertically to the one cell above it and to the four cells below it. Each cell within the fourth section connects vertically to the four cells above it and to the one cell below it.

For the interfaces between repository cells, lateral connections are established if two adjacent cells belong to the same section within the five-layer grid structure of the repository horizon. This rule is applied to both connections between columns and within columns (if local refining exists), but is limited to the cells within the repository horizon. For interfaces between a repository cell and a non-repository cell, the connection is built based on their relative elevations. The assignment of rock properties to repository cells is determined by the elevation of the cell and the corresponding hydrogeologic layer present at that elevation. To ensure proper connections between the repository cells (i.e., that they belong to the same emplacement drift), the user should align the columns properly by providing a correct order of data in the boundary file of the repository (see section 4.2.1). The first two pairs of the vertices define the orientation of the grid columns in the x-y plane.

3.4 Dual-permeability Grid

The dual-permeability gridding component of WinGridder generates numerical grids for fractured porous media consistent with the dual-continuum conceptual model. It can handle three types of fractured media:

(1) A single set of parallel, infinite fractures (Type #1) with uniform spacing within each hydrogeologic unit;

(2) Two orthogonal sets of parallel, infinite fractures (Type #2) with the same spacing within each hydrogeologic unit;

(3) Three orthogonal sets of parallel, infinite fractures (Type #3) with the same spacing within each hydrogeologic unit.

To generate a dual-permeability grid, WinGridder duplicates a primary single-continuum grid and adds inner connections between the corresponding fracture cells and matrix cells, with modified cell volumes and connection areas based on the fracture properties for each hydrogeologic unit. It uses the following formulas to calculate the volumes of the fracture cells and the matrix cells:

$$V_f = \phi_f \ V_n \qquad (3.1)$$

and

$$V_m = (1 - \phi_f)\ V_n \qquad (3.2)$$

where V_f and V_m are volumes of fracture and matrix elements, respectively, for the dual-permeability grid, and V_n is the volume of cell n of the primary mesh from which the dual-permeability grid is being generated. The porosity or volume fraction of fractures within the bulk rock is represented by ϕ_f.

The connection information for the dual-permeability grid is determined as follows:

Global fracture-fracture and matrix-matrix connection data are kept the same as those of the connections in the single-continua grid for the same/corresponding grid cell. This implies that permeabilities used for both fracture and matrix systems are the "continuum" values for both, relative to the bulk connecting areas. Inner connection distances between fractures and matrix within a primary grid cell are calculated as (Pruess, 1983):

$$D_f = 0 \qquad (3.3)$$

$$D_m = \frac{D}{6} \qquad \text{for Type \#1 fractures} \qquad (3.4)$$

$$D_m = \frac{D}{8} \qquad \text{for Type \#3 fractures} \qquad (3.5)$$

$$D_m = \frac{D}{10} \qquad \text{for Type \#3 fractures} \qquad (3.6)$$

and

$$D = \frac{1}{F} \qquad (3.7)$$

where D_f is the distance between the fracture center to the surface of matrix cells; D_m is the distance between the surface and an inner point of matrix cells, based on the quasi-steady state assumption (Warren and Root 1963); D is the fracture spacing; and F is the fracture frequency with the unit.

The interface area (A) between fractures and matrix cells is estimated by

$$A = A_{fm} \; V_n \qquad (3.8)$$

where A_{fm} is a volume-area factor, standing for total fracture-matrix interface areas per unit volume of rock, determined from the site fracture characterization studies.

4. WinGridder Architecture — An Overview

As a Windows-based software package, the execution order in WinGridder is determined by the user. Rather than present a flow chart mapping the overall structure of WinGridder, we present a framework of objects within the program. In terms of the software, objects may represent a real-world object (e.g., a fault or layer) or a grid unit (e.g., a cell or connection). Each object is described by a set of properties or methods, or both. Properties describe the relevant features of an object, while methods describe the behavior of the object. Knowledge of the important characteristics of real-world objects and their appropriate translation into the numerical world is a very important part of grid development.

WinGridder was written primarily in Visual Basic, following the principles of object-oriented programming. Some components, however, were written in C++ or other languages, while others are third-party software that should be properly installed on the user's PC before their components can be used (e.g., Notepad).

4.1. Graphic User Interfaces

Graphic user interfaces are major communication platforms between the user and WinGridder. These interfaces occur in the form of windows. The main window has many components that display information for the user or in response to the user's action (Figure 4.1-1). A caption bar located at the top of the main window shows the name of the current project (or a greeting message if no project open). A status bar located at the bottom of the main window shows the action status of WinGridder. The progress bar may tell the user how much of a current job has been done. The menu bars represent all functionality groups that the user can select. WinGridder also responds to certain key strokes, mouse actions, or some combination thereof in certain contexts. Within the main window, child windows are used to plot the grid or facilitate interactive grid design (Figure 4.1-2). Other dialog boxes, including the file selection box (Figure 4.1-3), the input box (Figure 4.1-4), and the printer setting box (Figure 4.1-5), automatically or manually engage to aid the user in various tasks. Through graphic user interfaces, WinGridder provides a user-friendly platform to generate variably complex 3-D grids.

4.2. Files

Files are the important communication media between WinGridder and the user, and also between the internal components of WinGridder. There are three groups of files used in WinGridder: input files, output files, and internal files.

4.2.1. Input files

Input files contain information (data) about objects in the real, or physical, world. The user should have all input data files prepared and formatted before using WinGridder to generate a grid.

Boundary Files: A boundary file contains location information about a 3-D object: its aerial extent (described by a polygon) and its top and bottom elevations. Each boundary file to be used in a project must contain, in its first line, a key word that is unique to that boundary file (Figure 4.2.1-1). Following the key word (on the next row) are two data entries, which represent the top and bottom elevations of the object. Each row thereafter consists of two numerical data entries: the x and y coordinates, respectively, of one vertex of the polygon. It is recommended that all vertices are listed counterclockwise. Note that in the repository boundary file (Figure 4.2.1-2), the first two pairs of the vertices define the orientation of the grid columns in the x-y plane for the repository.

The one exception to this format requirement for boundary files occurs when the corresponding object is a repository or other object that needs special treatment when gridding. In this case, the first four numerical data are A, B, C, and the thickness of the repository, respectively (Figure 4.2.1-2). Parameters A, B, and C are used to define the vertical position of the repository and are calculated with the elevation equation:

$$Z = A\,x + B\,y + C \qquad (4.1)$$

Equation (4.1) is used for easy interpolation in grid generation. If a repository cannot be described by a single plane equation, multiple plane equations should be used. In other words, the user should divide the repository area into several subrepositories such that each of the subrepositories can be described by a single plane equation. However, the current version of WinGridder does not support either multiple equations for one repository or multiple repositories.

Among all boundary files in one project, one file should describe the modeling domain boundary with the key word "DomainBound" in its first row of data. The key word for the repository files is "Repo." Examples of a domain boundary file and a repository boundary file are given in Figure 4.2.1-1 and Figure 4.2.1-2, respectively.

Layer Files: Hydrogeologic layers occupy the modeling domain and provide structure within the numerical grid generated by WinGridder. Information about hydrogeologic layers or stratigraphy is organized into several files: a "layer list" file, a "layer order" file, a "rock type" file, zone files (if any), and individual layer data files.

The "layer list" file contains a list of layer names (which are unique identifiers) and the data type (i.e., elevation or thickness; Figure 4.2.1-3). The first row of the "layer list" file contains the name of the "layer order" file.

The "layer order" file contains multiple entries, or sections, with each section of data representing one hydrogeologic layer and containing the layer name, a key word, the minimum x value, the minimum y value, the x-spacing, the y-spacing, the number of grid points in x-direction, and the number of grid points in y-direction (Figure 4.2.1-4). Note that the layer names have to be consistent with those in the "layer list" file. The order of the layer names has to reflect their relative stratigraphic positions. Three special key words, "top," "refer," and "bottom," are used for indicating three special horizons: the top boundary, the reference horizon, and the bottom boundary, respectively. For details about these three horizons, see Section 3.1 of this manual.

The "rock type" file lists all layer names and their corresponding rock names (Figure 4.2.1-5). Again, the names have to be consistent with those in the "layer list" file and in the "layer order" file, except the top layer whose name is "top" internally in WinGridder, and whose thickness is the difference between the top horizon and the top elevation of the second-from-top layer in the column. When a layer has multiple rock zones (e.g., tac#s in Figure 4.2.1-5), the rock name field in the file is filled by the name of a "zone" file that defines the rock zones (e.g., alch#s in Figure 4.2.1-5). The alphabetic characters "al" have to be the first two letters of the file name and are used to distinguish it from a common rock name.

A zone file starts with the number of zones and the rock name of the default zone (Figure 4.2.1-6). Several sections then follow, each defining one special zone. Each section starts with the number of vertices of the polygon and the rock name of the zone, and is followed by the x, y coordinates of the vertices. Each hydrogeologic layer should have one individual layer file containing either elevation or thickness data on a regular horizontal grid spacing (as described in the "layer order" file. All individual layer data files are in a binary format with a word length of eight bites for fast access. The data are ordered row-by-row (x-wise) and begin with the lowest x, y coordinate.

Fault Files: Each fault has its own fault file. If a fault branches into two or more faults, each branch off the main fault should be treated as a separate fault and have its own fault file. A fault file starts with a string that is the unique name of the fault. It then contains three sections of data that represent traces (x, y coordinates) of the fault at three different elevations. Each of the three sections starts with the number of points, NP, in the trace and the corresponding elevation, followed by NP (x, y) coordinates that define the trace at that elevation. Figure 4.2.1-7 shows an example of a fault file.

Drift Files: Each drift or other chain-type object (i.e., a line along which a user wants to arrange the nodes) has one drift file (Figure 4.2.1-8). Each drift file starts with a unique name and the drift radius, followed by x, y, z coordinates of the nodes that define the drift. If the name contains the key word “Closed,” the object is a closed chain, and WinGridder will connect the first node and the last node automatically. Otherwise, the object is an open chain. The key word “crosssection” is used to indicate that the drift is a dummy drift used for creating a cross-sectional grid. In the case of cross sections, the apparent dip of a fault, which is calculated based on the intersections between the fault and cross-sectional planes, will be assigned to the corresponding fault columns.

Borehole File: Only vertical boreholes are included here. Inclined or horizontal boreholes may be treated as drift-type objects. A borehole file contains data describing the x, y coordinates and unique name of each borehole to be located in the grid (Figure 4.2.1-9).

Fracture File: This file contains the fracture properties for each rock type and is used by the dual-permeability grid generator. Each row of data contains (from left to right) the five-character code name of the rock unit, fracture porosity, fracture aperture (m), fracture frequency (m^{-1}), and interface areas between fractures and matrix cells per unit bulk rock (m^2). The data format is (A5, 5X, 4E10.3).

4.2.2. Output Files

Output files provide information on the numerical grid in special formats as needed by users.

Whole Mesh File for TOUGH2: The entire mesh file for TOUGH2 is used as part of the input file for numerical simulations using TOUGH2. The mesh file contains three sections: the element section, the connection section, and the index section. In the element section, each row of data represents a cell, and the cells are ordered column by column and from top to bottom within a column. An example is shown in Figure 4.2.2-1. The mesh file extension is ".mesh."

Selected Mesh File for TOUGH2: This file contains a subset of the data from the whole mesh file for TOUGH2 and has the same format as the whole mesh file. The file extension is also ".mesh." A selected mesh file may represent a cross section through the model domain or a cluster of columns from the 3-D grid.

Selected Mesh File for Checking: This file is used to independently check the quality of the numerical grid generated by WinGridder, with particular emphasis on demonstrating consistency with the input hydrogeologic data. Each row of data contained in the file describes one cell, including its name, rock type, x, y coordinates, top and bottom elevations, the corresponding layer name, and the type of the cell (Figure 4.2.2-2). Overall, the data are ordered column-by-column and from top to bottom in each column. The file extension is ".mck."

Mesh File of Boreholes: This file is a special mesh file used for checking. It contains information about cells belonging to columns whose KIND property has the key word "bore." In other words, these cells belong to the columns that are designed to represent boreholes in the grid. The file format is the same as the Selected Mesh File for Checking, and the file extension is also ".mck."

List of Fault Columns: This file lists all fault column names within the grid. Each row of data has the column's name and the corresponding fault name (Figure 4.2.2-3). The file extension is ".lst."

List of Borehole Columns: This file lists all column names that are nearest to the horizontal positions of the boreholes provided in the borehole file, regardless of their designed KIND. Therefore, the columns in this file may be slightly different from those in the Mesh File of Boreholes in some cases (e.g., if a grid column was not specified to be centered at a certain borehole location during grid generation). The file extension is ".lst."

List of columns west of Solitario Canyon Fault: This file lists all column names located to the west of the Solitario Canyon Fault at Yucca Mountain (whose position is defined by a fault file). The file extension is ".lst." The

Solitario Canyon Fault can be replaced by any other delineation so long as the delineation is defined with the same format as a fault file and it separates the modeling domain into two parts (e.g., east and west).

List of Repository Cells: This file provides information about repository cells. It contains information about cells whose KIND property has the key word "InRepo" and whose volume is less than 10,000 m^3. Note that all cells within the five-section sublayering of repository have a KIND property of "InRepo." Also, no cells in a fault related column would have a KIND property of "InRepo" even though they may be located in the repository drift horizon. Finally, a repository cell may have volume greater than 10,000 m^3 if no refinement is specified in the gridding procedure. Each row of data in the file describes a cell, including its name, volume, KIND, corresponding layer, rock type, and elevation. Figure 4.2.2-4 shows an example. The file extension is ".mck."

List of Cells for a Specified Layer (including repository horizon): This file contains information on cells belonging to a user-specified layer and, in some cases, those cells lying close to that layer if the layer is pinched out at that location. The file extension is ".mck," and each row of data in the file describes a cell, including its name, volume, corresponding layer, rock type, elevation, and KIND (Figure 4.2.2-5). These cells make a full and single-cell-thick carpet over the domain. In other words, each column has one, and only one, representative cell in the file. If the layer is pinched out in a column, the cell immediately above is picked. If the layer has multiple cells in a column (e.g., the layer thickness exceeds the maximum thickness defined in the grid and was therefore subdivided), the topmost cell will be picked. If the user selects the repository horizon, the cells lying at the elevation of the repository drift are picked if they are located inside the repository area. For those cells outside the repository area, cell information is taken from UZ model layer tsw35 (Tptpll). If tsw35 is pinched-out in that column, the layer immediately above will be picked.

Segment File: This file contains geometry information about segments and is in a format used by Amesh (Figure 4.2.2-6). The file extension is ".grd."

Graphic Files: These files are figures generated by WinGridder and may be in variety of formats (e.g., EPS, BMP, etc.), depending on the configuration of the user's printer (to file). Examples of graphic files are a layout of columns (2-D map view of nodes and cell boundaries), a map view of lateral connections, and a cross-sectional view of the 3-D grid.

DualKOut file: This file lists the names of all cells in the dual-permeability grid and their counterparts in the primary single-continuum grid for checking purposes.

DualKElement file: This file contains "ELEME" data block for the dual-permeability grid readable by the TOUGH family of codes.

DualKConnection file: This file contains "CONNE" data block for the dual-permeability grid readable by the TOUGH family of codes.

4.2.3. Internal Files

Internal files are used by WinGridder to store information about a project of grid generation and to transfer data between its components. The user should be aware of the information contained in each internal file, but should not modify it externally because it may corrupt the project.

Project Files: This file contains information about columns, segments, name and path of the database files (if any), local refinement geometry (if any), and the current status of the project. The project file is saved when the user clicks Save on the Project menu and is loaded into memory when the user clicks Open on the Project menu. Using project files, the user can switch between different projects, modify a grid at a later date, or extract information from an existing grid without having to repeat each step. The file extension is ".prj."

Database Files: Because a numerical grid often has a large number of cells and connections, the information on the 3-D grid is stored in a database file for each project. Currently, the database file is in Microsoft Access format. There are three tables in the database file: Cells, Faces, and Sides. The Cells table contains cell configuration information in each record, while the Faces table records information about the connections. The Sides table is used to track all neighboring cells and the corresponding connections of every cell (except the top and bottom neighbors, which are in the Cells table). The purpose of using database files is to enhance the performance of WinGridder in handling larger grids. The file extension is ".mcd."

Node Files: The node file is an input file for the modified Amesh used to generate a primary 2-D grid. The user clicks the Save button on the design platform to save the node file. Important information contained within this file includes the names and the x, y coordinates of nodes. The file extension is ".in."

Cell Files: The cell file is an output of the 2-D grid generator component of WinGridder. The most important information in this file is the area of each column, which is loaded into memory by clicking Load→Grid under File in the menu bar. It has the same format as the cell file of Amesh. The file extension is ".cel."

Segment Files: This file is another output of the 2-D grid generator component of WinGridder and has the same format as the segment file. It contains geometry information about all segments that outline the layout of columns. The segment file is loaded by clicking Load→Grid under File in the menu bar. The file extension is ".grd."

Selected Columns and Segments: This file contains a list of columns and segments selected by the user. These columns and segments may define a profile that a user wants to plot or write out. The file extension is ".scc."

Files.2K (fixed name) file: This file lists all input files and output files used by the dual-permeability grid generator. These files are selected or specified by the user in the process of generating the dual-permeability grid with WinGridder (see Section 5.4.4).

Mesh files for DualK: These files include the mesh file and the connection index file of the single-continuum grid exclusively used by the dual-permeability grid generator.

4.3. Major Components

WinGridder is organized as an integrated set of program components. Each component may be used separately and in an arbitrary order, though components may be related to some degree. The following descriptions of major components provide a brief overview of WinGridder regarding its functionality. Although they are ordered as steps to generate a 3-D grid, users are encouraged to iterate these steps to design the most appropriate grid for their needs.

4.3.1. Loading Objects

This group of components is used mainly to read data from input files describing hydrogeologic and engineering objects. The order of executing individual components is determined by the user except for loading the domain boundary, which is always the first object to be loaded.

4.3.2. Assignment and Manipulation of Nodes

This group of components facilitates the assignment of nodes to hydrogeologic or engineering objects with user-specified density and style. The user can assign nodes for particular objects in any order, except for the base nodes (nodes for the domain) which have to be assigned first. Note, however, that the last nodes to be assigned may erase previous nodes if their locations are too close. Users can also add or delete nodes as well as manipulate the existing nodes by moving their positions and editing their properties. These components are the major tools for interactively designing the column layout, or map view, of the grid.

4.3.3 Generating the 2-D Grid

The major part of the 2-D grid generator in WinGridder is the modified Amesh code, a program written in C++. Its main functionality is to calculate a set of polygons that partition the whole domain space into columns using a set of given nodes by Voronoi tessellation techniques. Some modifications have been made to the original Amesh code (Haukwa 1998) so that it can handle the input and output files from the caller program, WinGridder.

4.3.4. Checking and Modifying the 2-D Grid

This group of components lets the user: (a) load grid files created by the 2-D generator, (b) plot the layout of columns and other 2-D grid information to the screen or to a printer, (c) remove undesired segments (or connections), and (d) move existing segments interactively.

4.3.5. Generating the 3-D Grid

This group of components consists of four parts: (a) building 3-D cells and corresponding vertical connections, (b) building lateral connections between columns, (c) building lateral connections within a column (if locally refined), and (d) generating a dual-permeability grid based on the above ECM grid. The user can choose to build "cells only", "lateral connections only", or both. WinGridder will also ask the user's input about some options of a 3-D grid.

4.3.6. Checking the 3-D Grid

This group of components allows a user to select and plot arbitrary cross-sectional views of the 3-D grid interactively. The user can, and should, use these components to visually check the quality of a 3-D grid.

4.3.7. Output

This group of components helps the user generate output files describing the results of a developed grid (referred to in Section 4.2.2).

4.4. Installing and Executing

4.4.1. Hardware and Software Requirements

WinGridder requires a PC and Windows 95/98/NT operating system to run. Some additional tools, such as TextEditor, that are useful for display and editing, require related third-party software to be loaded on the user's PC.

With this additional software, WinGridder serves only as a common interface. Without the related software, these additional tools will not function, but WinGridder will still function properly as a grid generator.

4.4.2. Setup

To install WinGridder, simply run SETUP.EXE from the distribution disk and then follow the instructions on the screen. After installation, unzip the example files that are distributed with the software to test for successful installation of WinGridder.

4.4.3. Executing

After WinGridder has been installed successfully, it can be run by clicking the START button and then PROGRAM on the Windows95/NT operating system. WinGridder will prompt the user to specify a new or existing project and which files should be loaded as domain boundary definition files. A greeting message "Welcome to using WinGridder!" will appear on the top of the screen.

To exit WinGridder, click the EXIT button on the Project menu. If the project has not been saved, WinGridder will ask the user if the project should be saved before exiting.

5. Functionality

5.1. Management of Projects

5.1.1. Creating a new directory/Getting in an existing directory

Purpose: To set the current directory. All input and output files for a particular project should be located in the same directory. Multiple projects can be put in the same directory, which is recommended for convenience if these projects share the same input files.

How:
1. Click SetDirectory or GetDirectory under Project on the menu bar;
2. Select the appropriate driver and directory from the dialogue box;
3. Click Ok.

If entering from SetDirectory, type the name of the new directory you want to make in the "Project Located at" box.

Tips: When selecting a directory, make sure it shows up in the "Project Located at" box. Otherwise, its parent directory may be selected as the current directory. The program will warn you about other errors (such as invalid directory names).

5.1.2. Opening an existing project

Purpose: To load information about an existing project. Information about a project is actually stored in two files. One is a text file (project file), which contains information on all columns (2-D cells), segments, sub-segments (if any), project status, etc. The other is an Access database file comprised of three tables with information about 3-D cells, connections, and the neighborhoods of each cell.

How:
1. On the menu bar, click Project → Open;
2. Select the appropriate project file;
3. Click Open.

Tips: All project files (files with extension ".prj") in the current directory are listed. Projects on other drivers or in other directories can be accessed using the mouse. File filtering can also be done by typing a different extension into the File Type box. Setting a correct directory as the current directory and always saving the project file with an extension ".prj" makes for easy tracking.

5.1.3. Saving a project

Purpose: To save a project and keep all related information on disk files to avoid recreating the project from scratch. When saving a 2-D project, no Access database file will be created.

How:
1. On the menu bar, click Project and then click either Save→2D or Save→3D;
2. Either click an existing project file on the file list or type a new file name in the file name box;
3. Click the Save button.

Tips: Always save a project file with the default extension ".prj." Always save a project as a 2-D project before generating a 3-D grid because the 2-D grid can be reloaded at any time as a basis for a 3-D grid with different options.

5.1.4. Close a project

Purpose: To clear from memory an opened project. This allows for initialization and generation of a new project.

How: On the menu bar, click Project → Close.

Tips: If quitting WinGridder, click Project → Exit instead of Project → Close.

5.2. Loading Input Files

5.2.1. Domain boundary

Purpose: To load the domain boundary file and generate an object that represents the model domain. Only the objects and processes within the domain boundary will be simulated. The data file describing the domain boundary should have the extension ".bnd." It is a text file that starts with the keyword "Domainbound," followed by the top elevation, the bottom elevation, and a series of (x, y) coordinates (see Figure 4.2.1-1).

How:

1. On the menu bar, click File → Load → Bound;
2. Select the appropriate file to be loaded.
3. Click Ok to load the selected files, or click Cancel to cancel the loading.

Tips:

1. Multiple files may be loaded at the same time. This is done in two ways: (1) by holding down the Ctrl Key while selecting each file needed (this allows for selection of multiple files that are separated by unwanted files in the list), and (2) by holding down the Shift Key while selecting multiple files in the list.
2. Multiple boundary files represent multiple areas requiring node assignment with different configurations. However, among the boundary files selected, only one file should have the keyword "Domainbound," which defines the simulation domain boundary. Furthermore, each file has to have a unique keyword or name. Any violations to these rules will result in failure to load, and WinGridder will issue a warning. At any time, the boundary files can be reloaded to update information regarding boundary objects, and all previous boundary objects will be cleared from memory. Note that when starting WinGridder, the program automatically asks for specification of the boundary files.

5.2.2. Repository

Purpose: To load the repository configuration files. A repository is a special kind of boundary object in WinGridder.

How: see Domain Boundary

Tips: Multiple repositories can exist in one domain as long as their names are unique and the first four letters are "Repo." Do not use the keyword "Repo" in any other files of "Boundary" objects unless that object is intended to be a repository. The interfaces of 3-D cells within the repository may not be located exactly at the interfaces of geological layers if they are not coincidental spatially, and the lateral connections may not be confined to within stratigraphic layers. Finally, regular four-sided columns will be aligned along the first segment in the polygon, while the other "Boundary" objects allow the user to choose the orientation of the node arrangement.

5.2.3. Faults

Purpose: To load fault data files.

How:
1. On the menu bar, click File → Load → Faults;
2. Select the appropriate file(s) to be loaded from the file list;
3. Click Open.

Tips:
1. The default extension of the fault files is ".fut." For details on selecting multiple files, see Tips in Section 5.2.1.
2. Fault files can be reloaded to update the fault object information, and all previous fault objects will be cleared from memory. Do this to include a different number of faults than used in the previous iteration of a grid. When exiting WinGridder, all fault objects are cleared from memory.

5.2.4. GeoLayers

Purpose: To load data describing stratigraphic structures and rock type distribution.

How: 1. On the menu bar, click File → Load → GeoLayers;

2. Select the appropriate layer list file and click Open;

3. Select the appropriate rock type file and click Open;

4. WinGridder asks the user if they want to have a higher water table west of the Solitario Canyon fault (a special situation at the Yucca Mountain site). Click Yes if you want WinGridder to do so, and it will provide an input box to enter the elevation difference in meters. WinGridder then prompts the user to select a fault from a list of fault files across which the water table elevation step will occur.

Tips: 1. The default extensions of the layer list file, the layer order file, and the rock type file are ".txt," ".dat," and ".roc," respectively. Regarding water table configuration, in most cases the user will provide an x, y, z data file describing the fixed position of the water table. When asked by WinGridder if a higher water table should be used for a portion of the grid, the user simply clicks No. On the other hand, if the user wishes to evoke this option, he or she can select another fault file instead of the Solitario Canyon fault file to represent such a characteristic line. WinGridder first forms a polygon with the trace data of the selected fault and the northwest and southwest corners of the domain boundary, and then marks all columns within this polygon, which will have a higher water table in generating 3-D cells.

2. The user can reload the layer files to update the stratigraphic information, and all previous layer objects will be cleared from memory. When exiting WinGridder, all layer objects are cleared from memory.

5.2.5. Drifts

Purpose: To load files describing the layout of drifts and other line objects.

How: 1. On the menu bar, click File → Load → Drifts;

2. Select the appropriate drift files from the list;

3. Click Open.

Tips: 1. The default extension of the line object files is ".dft" or ".rvr." Any line object can be formulated into this category. Examples are rivers, special rock bodies that have line-type shapes, borehole paths, or any line area requiring a different grid density from surrounding areas. For instance, a "dummy" line object is very useful for generating a grid for a cross-sectional profile. For details about how to select multiple files, see Tips in Section 5.2.1.

2. Drift files can be reloaded to update the information of the drift objects and all previous drift objects will be cleared from memory. When exiting WinGridder, all drift objects are cleared from memory.

5.2.6. Boreholes

Purpose: To load location information about vertical boreholes.

How: 1. On the menu bar, click File → Load → Boreholes;

2. Select an appropriate borehole file;

3. Click Open.

Tips: 1. Borehole data are used for the assignment of nodes at exact coordinates corresponding to borehole locations in map view. Note that assignment of borehole nodes may conflict with the requirements of other special objects such as repositories and faults.

2. The borehole file can be reloaded to update the information about borehole locations, and all previous borehole objects will be cleared from memory. When exiting WinGridder, all borehole objects are cleared from memory.

5.3. Column Design and 2-D Grid Generation (map view)

5.3.1. Getting in Design mode

Purpose: To prepare the working platform for column design and map-view grid generation.

How: 1. On the menu bar, click Design → 2D, and a design form will show up on the screen;

2. Click the Display button located on the right side of the window to display a current view on the plotting board (the plotting board is located on the left side of the plot window). The domain boundary and outlines of other bound objects will be drawn on the board.

Tips: In design mode, the user cannot print a graph to a printer or a file. Printing can be done, however, by switching to plot mode (i.e., by clicking the Plot button on the menu bar).

5.3.2. Assigning nodes for objects

Purpose: To assign nodes to represent each object. Assignment of nodes is the first step of column design. These nodes will be the basis for the column layout of a grid.

How: 1. Select DomainBound from the pull-down list of the available objects located in the top-right corner of the design window. The domain boundary will be highlighted by a thick line on the plotting board.

2. Specify the average distance between neighboring nodes in the Average Distance box.

3. Click the Add Nodes button. WinGridder will ask the user to specify: (a) the angle between node orientation and the positive x-axis, (b) the ratio of y-spacing over x-spacing, and (c) whether or not a rectangular type of node arrangement should be used. WinGridder then generates nodes and displays them on the plotting board as red dots. The current (total) number of nodes is shown on the status bar.

4. If the nodal array is unsatisfactory, click the Clean or Clean All button to delete nodes, and return to Step 2 to reassign nodes. Otherwise, continue. The Clean button deletes only those nodes that represent the object currently selected, while the Clean All button deletes all existing nodes.

5. To assign nodes to remaining objects within the domain boundary, select another object from the "Select and Object" pull-down list and follow Steps (2) through (4) above.

Tips: 1. The Clean All button deletes all existing nodes; therefore, nodes for the domain boundary must be reassigned before nodes for any other objects may be assigned. If the user wishes to delete only a few nodes, it should be done interactively.

2. There is no special order, or sequence, requirement for assigning nodes to objects other than the domain boundary, which must be done first. Keep in mind, however, that nodes assigned later may delete the previous existing nodes if the nodes are located in close proximity.

3. A node's KIND property determines which object it represents. The KIND property is usually the name of the object that the node represents and can be later modified interactively. If a node is shared by multiple objects, its KIND property is the combination of the names of those objects.

4. A repository is a special type of polyhedral object. Node orientation within a repository object is defined by the first two vertices.

5. A fault is represented by three parallel rows of nodes. WinGridder displays a fault with different colors: yellow for the fault nodes and green for the side nodes. A side node's KIND property usually contains a keyword which consists of the fault node's name and either "Side0" or "Side1," depending on which side of the fault it is located. This designation allows a fault to be represented as a nonvertical column in a 3-D grid.

6. A special "group object" exists for selecting similar objects and assigning nodes to them collectively instead of on an individual basis. For example, the user can select AllFaults instead of each fault individually. The order of node assignment is the same as that for loading objects. The group object for boundary objects is called Other Area Object.

7. Selecting SingleNode from the "Select and Object" pull-down list allow the user to add a node Interactively by using the mouse and keyboard. For information about adding nodes interactively, see Section 5.3.3.

8. Selecting FreeHandObjects from the "Select and Object" pull-down list allows the user to add several geometry objects, such as a line, a circle, and a polygon, interactively on the grid for plotting purposes only. Assigning nodes to these interactively added objects is not permitted in the current version of WinGridder.

When the Average Distance box is active, pressing the Enter (or Return) key is equivalent to clicking the Add Nodes button.

5.3.3. Interactively add nodes

Purpose: To add a node at an arbitrary location using the mouse and keyboard. This function complements the function of adding nodes for objects. It provides some flexibility for a user to design a column layout more suitable for representing complex simulation objects, especially when the objects overlap each other in space.

How: 1. Select SingleNode from the "Select and Object" pull-down list and click the Add Nodes button;

2. Point the mouse (the mouse occurs as a "+" on the screen) to the location you want to place the node, the current x, y coordinates will be shown on the status bar of the main window;

3. Click the left button on the mouse while holding down the Ctrl-key on the keyboard.

4. You may also edit the KIND property assigned to a node (see Section 5.3.8).

Tips: 1. Use Zoom to get an enlarged view of the layout. If an interactively added node is not satisfactorily located, it may be modified by moving it or by deleting it using related functionality. If the mouse is positioned outside the domain boundary, no node will be added because all valid nodes have to reside within the domain boundary.

2. If an added node is too close to an existing node, WinGridder will post a warning. The minimum distance criterion can be changed by typing a new value in the Average Distance box on the plot form. However, the use of very small minimum distance values is not recommended because the 2-D grid generator will generate columns for all nodes, even if some of them exactly overlap each other. Tracking such an error may be difficult and very time-consuming.

5.3.4. Select a node

Purpose: To select a node as the current node.

How: 1. Point the mouse at the node to select;

2. Click the left button on the mouse while depressing the shift key. The selected node appears in blue.

Tips: Sometimes, a red line or box appears on the screen after a node has been selected. This is because moving the mouse with both the left button and the shift key down creates a durable red line on the graph. To erase this line, just draw another line on the screen (do not press the shift key) and click the Display button.

5.3.5. Clean nodes

Purpose: To delete unwanted nodes.

How: Method 1. Click the Clean button. WinGridder will delete all nodes that represent the object selected.

Method 2. Click the Clean All button. WinGridder will delete all existing nodes.

Method 3. Draw a square that encloses all nodes to be deleted (see Zoom:Method 1 for how to draw a square on a graph) ; Press Delete on the keyboard.

Tips: 1. Method 2 actually initializes your design work.

2. When cleaning nodes, make sure neither the Grid box nor the Connection box is checked or an error message will appear.

5.3.6. Move nodes to other locations

Purpose: To relocate an existing node.

How: 1. Select a node to be moved;

2. Point the mouse to the new location where the node should reside (see the status bar on the main window for new x, y coordinates);

3. Press the M-key on the keyboard. WinGridder will show the node at the new location.

Tips: After a node is selected, it is in current status. Therefore, it can be moved multiple times without having to re-select it.

5.3.7. Viewing a node's Name, LBNL name, and KIND

Purpose: To view a node's Name, LBNLname, and KIND properties. A node's Name is the internal ID for the node (or the column in the final 3-D grid), while the LBNLname is a conventional coded name for a column used by the LBNL UZ modeling group. For example, if the first letter of a LBNLname is upper case, this node represents a fault column. A node's KIND property indicates to which object a node belongs.

How: 1. Point the mouse at the node for checking;

2. Press right button down to see the three properties. The first row of information contains the LBNLname with the node's Name (internal ID) in parentheses, while the second row indicates the KIND property.

Tips: The node selected by WinGridder is the one that is nearest to the mouse's location.

5.3.8. Modifying the KIND of a node

Purpose: To assign or edit the KIND property of a node. The KIND property indicates which object a node belongs to and determines the kind of column to be generated for the node.

How: 1. Point the mouse at the node to be edited;

2. Click the right button on the mouse while holding the Ctrl-key down. A text box appears for editing the property;

3. Edit the KIND property;

4. Press the Enter key to end the editing.

Tips: Be careful when modifying a node's KIND property to "Fault" because the whole column of cells generated for that node will be assigned fault rock properties.

5.3.9. Save node information to a disk file

Purpose: To save a node file as input for the 2-D grid generator. Only x, y coordinates of nodes will be used in the 2-D grid generator, and the Names will be passed to the output files of the 2-D grid generator. The node file contains information about the domain boundary, and its default extension is ".in."

How: 1. Click the Save button on the plot form;

2. Select a file from the list of existing node files; or

3. Type a new file name in the file name box;

4. Click Save to finish.

Tips: Selecting an existing file results in overwriting the file with new node information.

Warning: Not all node parameters will be saved in the node file. If you want to keep all parameters in a disk file, please use the saving project functionality (see Section 5.1.3).

5.3.10. Generating a 2-D grid

Purpose: To calculate a 2-D grid (layout of columns) for a given node file using Voronoi tessellation techniques. The results are written to a segment file and a cell file.

How: 1. On the menu bar, click Generate → 2-D;

2. Select a proper node file (".in") from a list of existing node files;

3. Click Open to initiate the work. A DOS window will show until the job is done.

Tips: 1. Regenerate a 2-D grid whenever nodes have been added or deleted. Note that any unsaved modifications to the segments or connections will be lost after grid regeneration.

2. WinGridder may post a warning if it cannot find the executable file for the 2-D grid generator. If this is the case, check to see that the WinGridder software has been properly installed on your PC. The executable file for the 2-D grid generator is the modified Amesh code compiled by MS Developer Studio version 5.0 (written in C++). Note that the original version of Amesh will not work here.

5.3.11. Loading a 2-D grid into memory

Purpose: To load the primary 2-D grid. WinGridder first reads the information about segments and places this information into a segment collection (an internal object used as a warehouse to store all segments), and then updates the information describing the area and the neighborhoods of all columns.

How: 1. On the menu bar, click File → Load → Grid;

2. WinGridder will post a warning if a grid file already exists in memory;

3. Select an appropriate segment file (extension ".grd") from a list of available segment files;

4. Click Open to initialize loading; WinGridder will specify how many segments have been loaded.

Tips: 1. After loading the 2-D grid into memory, modifications may be made to the segments or connections in the 2-D grid, as necessary. Remember to save and regenerate the modified 2-D grid before continuing on to 3-D grid generation, or the modifications will be lost. Another way to access a 2-D grid is to open the corresponding 2-D project, which should be saved after completion of the 2-D grid design (column scheme).

2. Before loading a new grid into memory, any existing grids in memory must first be unloaded. To do this, go to File→Unload→Grid.

5.3.12. Select a connection or a segment

Purpose: To select a segment as the current one. In the column layout, selecting a segment is equivalent to selecting a connection. A segment may be either moved or deleted after it is selected.

How: Select two adjacent nodes consecutively. The color of the selected nodes will change to blue, indicating that the segment between these nodes is selected.

Tips: To see if two nodes are connected, check the Connection box in the Design window and click the Display button. If two selected nodes are not connected, no segment is selected. WinGridder only remembers the last two selected nodes.

5.3.13. Interactively delete unwanted segments

Purpose: To delete unwanted segments interactively to enforce rules regarding the connections within a fault, repository, or other object. After removing a segment, all columns and segments that are affected will be recalculated and updated accordingly.

How: 1. Select a segment;

2. Press the Delete key while holding the Shift key down. If the selected segment does not exist (e.g., two selected nodes are not connected), the warning message “Can’t remove this segment!” will appear.

Tips: A segment cannot be removed if it is either:

(1) A part of the domain boundary; or

(2) A segment whose neighboring column only has three (lateral) sides.

If you try to delete one of these types of segments, WinGridder will issue a warning and the segment will not be deleted.

5.3.14. Interactively moving a segment

Purpose: To change the relative position of a segment between two nodes. The Voronoi tessellation used in Amesh (the 2-D grid generator) always places a segment midway between two neighboring nodes. In some situations, however, this may not be desirable. For example, a user may want a node located in the center of the column. After moving a segment, all affected columns and segments will be recalculated and updated accordingly. The selected segment only shifts a given distance along the connection line between two neighboring nodes without changing its length. All other segments connected to the selected segment will be adjusted accordingly by WinGridder, to preserve the connections. Thus, the affected columns may not be limited to the two neighbors of the selected segment.

How: 1. Select a segment;

2. Click Edit on the menu bar and then click Move Segment;

3. Type a relative distance into the input box;

4. Click OK and WinGridder will perform all related calculations.

Tips: The relative distance is defined as a fraction of the distance between two neighboring nodes, while the sign ("+" or "-") indicates the direction of movement.

5.3.15. Adding other objects (lines, circles, and polygons)

Purpose: To add objects interactively for special node assignments or for simply drawing geometry objects on the graph.

How: 1. Select FreeHandObjects from the "Select and Object" pull-down list;

2. Click Insert;

3. Select an object type and assign a name and drawing-width (or radii for a circle) to the object on the new object properties form.

4. Click Ok to start creating the new object. WinGridder will provide instruction about how to create an object of the specific type selected.

There are three types of objects you can create:

(1) *A line:* Point the mouse to the starting position of a line;

Press the space bar to start creating;

Move the mouse with left button down to draw a line;

Press the space bar to confirm the end of a segment;

Click Insert to modify the width of the line at any time;

Press the Enter key to confirm the end of the line.

(2) *A circle:* Point the mouse to the location where the circle should be placed;

Press the space bar until a circle appears on the board;

Click Insert to modify the radius of the circle as needed;

Press the Enter key to accept the circle.

(3) *A polygon:* Point the mouse to the starting position of the polygon;

Press the space bar to start creating;

Move the mouse with left button down to draw a line;

Press the space bar to confirm the end of a segment, WinGridder will add the last segment automatically.

Press the Enter key to accept the polygon.

Tips: The current version of WinGridder allows these objects to be created as graphic objects only (i.e., WinGridder does not allow these objects to be assigned nodes). If such objects require the assignment of nodes, proper input files should be created for them.

5.3.16. Deleting unqualified segments

Purpose: To delete any unqualified segments (i.e., the fault related segments that do not meet the requirements described in Section 3.2, or the segments that are too short compared to the areas of their neighboring columns).

How: 1. On the menu bar, click Design → Remove bad Segments;

2. Type an appropriate minimum ratio of the segment length over the column area;

3. WinGridder will post a warning and abort removing segments if it finds an invalid segment; otherwise, it states how many unqualified segments have been removed.

Tips: 1. The ratio of the segment length over the column area is calculated as the square of the segment l ength divided by the area of the adjacent column.

2. Invalid segments are most likely fault-related segments. If so, first check the arrangement of fault-related nodes; specifically, whether or not the KIND properties of side nodes match the corresponding fault nodes. WinGridder announces an error if it finds either of following two conditions:

(1) A segment with zero length;

(2) A column whose area is less or equal to zero.

5.4. Generating 3-D Grid

Once the 2-D grid (map view) has been prepared, the user can generate the 3-D grid with several options. Generating a 3-D grid may require substantial CPU time, especially if the grid contains a large number of cells and connections. Therefore, WinGridder allows the user to generate different components of the grid individually, if necessary.

5.4.1. Building cells and vertical connections only

Purpose: To generate cells and corresponding vertical connections for a given layout of columns and segments.

How: 1. On the menu bar, click Generate → 3-D → Cells/Vertical Cons Only;

2. Several pop-up windows appear:

(1) Specify whether or not to save the project automatically;

If yes, select an existing project file or create a new project file;

Click Save, and WinGridder will save generated cells and connections to the database during building and other project information to the project file after building.

If no, WinGridder will store all information in memory until the user saves the project by clicking Save on the Project menu bar.

(2) Specify whether or not to use the long LBNLname;

If yes, LBNLnames of the cells can be as long as eight characters. Otherwise, only five characters are used. Because the LBNLname has to be unique, as required by the TOUGH2 code, the longer LBNLname should be used for grids having a very large numbers of cells.

(3) Enter the minimum thickness of a cell. For a given column, any layer having a thickness smaller than this value will be ignored, and no cell will be generated to represent it.

(4) Enter the maximum thickness of a cell. For a given column, any layer having a thickness larger than this value will be represented by multiple cells with equal thickness.

(5) Specify whether or not to have locally dense gridding within the repository area. If yes, local refinement will be applied inside the repository.

WinGridder shows the progress of the building process through the progress bar and the status bar on the main window. WinGridder first builds cells for all fault-related columns, and then builds cells for the other columns.

Tips: 1. If the grid involves a large number of cells, use the option of automatically saving the project to enhance the performance of WinGridder.

2. If any required information is not ready (e.g., no layer data exist), WinGridder will post a warning before generating the 3-D grid. If this is the case, click Ok and follow the instructions on the screen to load the necessary information.

5.4.2. Building lateral connections only

Purpose: To create lateral connections (except top/bottom connections) once the 3-D cells are created. The lateral connections are created segment by segment according to the rules described in Section 3.

How: 1. On the menu bar, click Generate → 3-D → Lateral Cons Only;

2. Two pop-up windows appear and the user must specify:

(1) Whether or not to laterally connect Top Cells or Bottom Cells even when the neighboring column does not have a cell of the same layer. This is designed to deal with special cases in which certain layers do not exist in one of two neighboring columns because of either surface erosion, layer pinchouts, or layers occurring below the bottom boundary (e.g., the water table). If the user answers yes, WinGridder will laterally connect all Top Cells and laterally connect all Bottom Cells in adjacent columns, even if they do not belong to the same layer. If the user answers no, those connections will not be created.

(2) The minimum thickness of 3-D cells: This value is used when a lateral connection is created for two cells belonging to different layers (i.e., where a layer pinches out). In this case, 10% of the specified minimum thickness is used as the height of the interface (connection) between the two cells.

Tips: 1. This functionality always saves the project automatically and should be used only after the 3-D cells have been successfully generated. If the 3-D cells have not already been generated, WinGridder will post a warning and abort the process.

2. The value of minimum cell thickness provided here (for lateral connections) has no effect on the minimum cell thickness specified in Section 5.4.1 (for building 3-D cells and vertical connections).

5.4.3. Building the full 3-D mesh

Purpose: To create a 3-D grid for a given layout of columns and segments (2-D grid) with related information.

How: 1. On the menu bar, click Generate → 3-D → Full Mesh;

2. Several pop-up windows appear and the user must specify:

(1) Whether or not to save the project automatically. If yes, select an existing project file or create a new project file (this saves the generated cells and connections to the database during building and the other project information to the project file after building). If no, WinGridder will store all information in memory until the user saves the project by clicking the Save on Project menu.

(2) Whether or not to laterally connect Top Cells or Bottom Cells anyway, even though the neighboring column does not have a cell of the same layer. This is designed to deal with special cases in which certain layers do not exist in one of two neighboring columns because of either surface erosion, layer pinchouts, or layers occurring below the bottom boundary (e.g., the water table). If the user answers yes, WinGridder will laterally connect all Top Cells and laterally connect all Bottom Cells in adjacent columns, even if they do not belong to the same layer. If the user answers no, those connections will not be created.

(3) Whether or not to use the long LBNLname. If yes, the LBNLnames of cells can be as long as eight characters. Otherwise, only five characters are used.

(4) The minimum thickness of a cell. For a given column, any layer having a thickness smaller than this value will be ignored, and no cell will be generated to represent it.

(5) The maximum thickness of a cell. For a given column, any layer having a thickness larger than this value will be represented by multiple cells with equal thickness.

(6) The maximum thickness of a cell below the repository level. For a given column, a layer that is below the repository (or other user-specified elevation) and has a thickness larger than this value will be represented by multiple cells with equal thickness.

(7) Whether or not to have a locally dense grid within the repository. If yes, local refinement will be applied inside the repository.

Tips: This function is a combination of “Building 3-D cells only” and “Building lateral connections only”. With this function, WinGridder performs both jobs (in sequence) automatically. Note that this may require substantial CPU time. Therefore, it is important to make sure that the 2-D grid, or the layout of columns and segments, is satisfactory before generating a 3-D grid.

5.4.4 Building a dual permeability mesh

Purpose: To build a dual-permeability (dual-k) grid for representing both fracture and matrix continua.

How: 1. On the menu bar, click Generate → 3-D → Dual-K mesh;

2. Several pop-up windows appear and the user must:

(1) Select an appropriate file that defines the fracture properties of each rock layer;

(2) Specify whether or not an ECM mesh has been saved for dual-k grid generation. If yes, select a proper “m2k” file. Otherwise, click No and enter the name of the “m2k” file.

Tips: The single-continuum mesh (Effective Continuum Mesh, ECM) must be in a format readable by WinGridder’s dual-permeability grid generator and must be saved for the dual-k grid generator instead of for TOUGH2.

5.5. Generating a 2-D cross-sectional grid or a 1-D grid

Purpose: To create a vertical cross-sectional grid or a single-column, 1-D grid. These are special cases of a 3-D grid.

How: 1. Create data files of a line object (i.e., a dummy drift) and an appropriate domain boundary. The line object can be any shape.

2. Create a 2-D grid (map view) of a single row of columns for the line object;

3. Build a 3-D grid based on the above column layout.

Tips: 1. If the local refinement is applied, the 2-D cross-sectional grid is actually a 3-D grid.

2. Building a 3-D grid for a single column will result in a 1-D grid.

3. Two-dimensional grids and 1-D grids can be extracted as subgrids from a full 3-D grid.

5.6. Rock type assignment

Purpose: To assign each cell with an appropriate rock name (i.e., material assignment) according to the rock-layer information provided. Fault cells will have a rock name consisting of the first three letters of the rock name of the corresponding layer and the first two letters of the fault name.

How: 1. On the menu bar, click Edit → Assign Properties.

2. If the layer information is not ready, WinGridder will post a warning and ask the user to load the layer information.

Tips: Rock type assignment is usually done when the 3-D grid is created. This functionality is designed to offer the user an opportunity to modify rock properties without regenerating the 3-D grid, which can be time consuming.

5.7. LBNL Cell-Naming

Purpose: To assign each cell with a unique LBNL cell name and each column a unique LBNL column name. The LBNL names are coded names used in the LBNL UZ modeling group. An LBNL column name consists of a letter and two digits. If the letter is capitalized, the column is a fault column. An LBNL cell name is created by adding two (or four, for the long LBNL name option) letters from a layer name (an LBNL layer name) before the column name. If multiple cells exist within a column for a single layer (i.e., the layer's thickness exceeds the user-specified maximum cell thickness and has therefore been split into multiple cells), the last letter of the letters representing the layer will be replaced by ordered ascii characters. The first character of the long LBNL name is empty and will be used to get dual-permeability grid.

How: 1. On the menu bar, click Edit → LBNL Naming;

2. If the layer information is not ready, WinGridder will post a warning and ask the user to first load the layer information. If the name length has not been assigned, WinGridder will ask for specification.

Tips: The LBNL names are assigned when a 3-D grid is generated. However, this functionality can be used to modify these names (e.g., switch from a long name to a short name). If 3-D cells are not ready, WinGridder will post a warning and abort the process.

5.8 Plotting Graphs

5.8.1 Getting into plot mode

Purpose: To make WinGridder ready for plotting graphs.

How: Click Plot on the menu bar. A plot form will appear on the screen. The default plotting plane is map view.

Tips: Nodes cannot be assigned to objects in plot mode. To assign nodes to objects, switch to design mode by clicking Design on the menu bar.

5.8.2. Select a view

Purpose: To select either a map view or cross sectional view.

How: Pull down the list of views from the "Select a plot plane" window located in the upper right corner of the plot form, and select one of them.

Tips: The default view in plot mode is map view. To view a grid in cross section, select a row of columns or draw a line on the map view.

5.8.3. Select or unselect an object to view

Purpose: To help users plot graphs with different combinations of objects.

How: Click the check boxes corresponding to the objects intended for display. An object may be either a grid object (i.e., Cells) or a geologic object (i.e., Faults).

Tips: WinGridder automatically checks the availability of objects, and only the available objects will be activated for selection. The domain boundary and other area objects (e.g., a repository) will always be shown on the graph if they are available.

5.8.4. Map view

Purpose: To show a map view of a grid object or geologic object on the screen.

How:
1. Select Map view from the "Select a plot plane" pull-down list;
2. Select all objects to be included in the graph;
3. Click Display. A graph will be shown on the left part of the plot form.

Tips: Modify the graph by selecting different objects, and then click the Display button. Different parts of the graph may be plotted, with different scales, by using the Zoom or Relocating functions.

5.8.5. Zoom

Purpose: To plot a graph at different scales.

How: Method 1: Move the mouse to a corner of the rectangle that you want to show;

Press the left button of the mouse;

Move the mouse to the opposite corner of the rectangle with the left button down; Release the button and a rectangle will be drawn on the graph;

Click Zoom button and a zoomed graph will be plotted on the plotting board.

Method 2: Click Zoom button until a dialog window occurs;

Type the coordinates of the left-bottom and the right-top corners of the plotting rectangle into the proper cells on the spreadsheet to zoom the graph into your specified dimensions, or click Default button to return to the default zoom status;

Click the Ok button. A zoomed graph will appear on the plotting board.

Tips: You can zoom a graph of map view regardless of the mode (i.e., Design mode, Plot mode, or Select mode). However, Method 1 of the zoom function is not available if a graph is a cross-sectional view.

5.8.6. Relocating the center of a view

Purpose: To plot a different part of the graph with the same scale.

How: Method 1: Point the mouse to a location that is the new center you want;

Double click the left button. A new view of the graph will occur.

Method 2: Press one of the Arrow keys on your keyboard. Each press will shift the view by 20% to the direction of the Arrow key you pressed.

Tips: You can relocate the center of a view regardless of the mode (i.e., Design mode, Plot mode, or Select mode). However, this function is not available if a graph is a cross section view.

5.8.7. Drawing a straight line on a graph

Purpose: To add a straight line to a graph. The line may indicate the location of a profile.

How: 1. Point the mouse to the starting position of the line;

2. Move the mouse with both the left button and the Shift key down.

Tips: To erase the line, move the mouse with the left button down only and then click the Display button.

5.8.8. Adding, Deleting, Moving a label and editing its font

Purpose: To add a text label at the specific location on a graph with various fonts.

How: ***(1) Adding***: Click Add Label(s) button. A dialog box will appear to ask you if you want to add labels automatically (if it is the first label to add). If you click Yes, WinGridder will create labels for each objects loaded. All the created labels are located at the top left corner of the plotting board. You can move these labels one by one to arrange them properly. Otherwise, WinGridder will ask you to type the text you want the label to have.

(2) Moving: Point the mouse to a label;

Press the left button down and hold;

Move the mouse to the location where you want to put the label;

Release the left button.

(3) Deleting: Move a label to the lower-left corner of the graph (the area below the x-axis and left of the y-axis).

(4) Editing font: Point the mouse to the label;

Click the right button;

Select proper color, size, style, and other font factors on the font edit board.

Tips: WinGridder is designed to plot primary graphs of the grid. You may add labels to indicate any objects on a graph. However, for a final graph, you should save the graph to an EPS file and then use other commercial graphic software to modify the graph.

5.8.9. Interactively select columns (connected or not)

Purpose: To select a subset of a grid.

How: 1. Click File menu bar and point to Save and then SelectedMesh; Click Ok button on the warning box;

2. Click No button on Question box, which then asks you if the selected columns are already in a file;

3. Click the Pick Cells button on the plot form. The button becomes the Picking Cells button which indicates that the WinGridder now is in Picking Cell mode. You can follow these steps for three different cases:

(1) Select a profile**:** Select nodes one by one in an order that traces a profile;

(2) Select boreholes**:** Select nodes that represent boreholes. If two nodes selected in sequence are not connected, WinGridder will warn you that the two columns are not connected and a red line is drawn between these two columns instead of a blue line. Just click Ok button on the warning box to continue.

(3) Select an area**:** Select nodes one by one in an order that traces a boundary of the area.

4. Click the Picking Cells button to finish. The Picking Cells button changes to Pick Cells button which means you have completed the selection of columns. The Selected Cells check box will be activated.

Tips: 1. Selecting an area is very similar to selecting a profile. Each consists of a chain of connected columns that requires that the consecutively selected columns be adjoined. The only difference is that the chain that encloses an area is a closed polygon, while the chain of a profile is open. In other words, WinGridder decides whether you have selected an area or a profile by checking if two end columns of the chain are connected. If so, an area is selected, and WinGridder will search all columns within the polygon defined by the chain and have them selected, too. Otherwise, only a profile is selected.

2. By selecting columns, all segments that are interfaces between the selected columns are automatically selected. Hence, this functionality actually facilitates a user to select a subgrid.

3. If all selected columns are isolated columns (no connections between them), no segment will be selected, and the selected subgrid is a set of multiple 1D grids.

4. If the selected columns exist, clicking Save SelectedMesh in the File menu will cause WinGridder to show a saving file interface for you instead of the warning box of no columns selected. In this case, you have to deselect the selected columns as described below before you can select new subgrids.

5.8.10. Deselecting columns

Purpose: To discard the selection of a sub grid.

How: 1. Click Pick Cells button, the Pick Cells button becomes Picking Cells button;

2. Click Picking Cells button, WinGridder will inform you that the selection has been cleared.

Tips: 1. Each time you click the Pick Cells button, the previous selection will be cleared. Thus, clicking Picking Cells button without selecting any columns will result in an empty collection of the selected columns.

2. Note that deselecting columns before leaving the select mode to other mode (e.g., plot mode or design mode) is very important. Otherwise, you may not be able to return to the selecting mode.

3. Save the selection for next time before discarding it (see Section 5.9.2).

5.8.11. Show selected subgrid (map view)

Purpose: To visualize the selected subgrid with different colors.

How: 1. Check the check box of Selected Cells;

2. Click Display button. The selected subgrid will occur with a different color.

Tips: If the check box of Selected Cells is not activated, no column has been selected. You have to select the columns before you can show them.

5.8.12. Show a cross-sectional view of a selected profile

Purpose: To plot a cross section view of a 3-D grid.

How: 1. Select a profile;

2. Show it in a map view (see Section 5.8.11);

3. Select Cross section view from the "Select a plot plane" list;

4. Click Ok button on a message box showing how many columns you have selected;

5. Click Display button.

Tips: You can choose components to show by checking them. However, some components cannot be shown on a cross-sectional view. Fault cells are always yellow.

5.8.13. Show a cross-sectional view based on a straight line drawn by a user

Purpose: To visualize a 3-D grid on a cross section view.

How: 1. Draw a straight line on the map view of a grid;

2. Select "Cross section" view from the "Select a plot plane" list;

3. Click Ok button on a message box showing how many columns you have selected;

4. Click Display button.

Tips: This functionality is very similar with the one above. However, drawing a line to define a profile is much easier than selecting a chain of columns. The disadvantage is that two connected neighboring columns may be separated by another column in plotting if the line just passes a corner of that column.

5.8.14. Printing graphs

Purpose: To get hard copies of a graph.

How: 1. Click Print button;

2. Select a printer or modify settings of a printer if needed on the printer dialog box.

Tips: Clicking the Print button will send the current graph on the plot board to a printer. WinGridder can use any printers that are available in the Windows operation environment of your PC. Note that if you select a printer other than the Windows' default printer, the selected printer will become the Windows' default printer after printing of the graph. Any modifications you made to the settings of the printer will be permanent. Of course, you can return to the Windows' original default printer through Windows operation system (i.e., the printer setting dialog box).

5.8.15. Printing a graph to a disk file (graphic files)

Purpose: To get an electronic copy of a graph that can be used by other graphical software.

How: Same as printing a graph on a printer, but select the File that is a virtual printer installed through the Windows' setting printers functionality. WinGridder will ask you to input a file name.

Tips: To use this functionality, you have to install a virtual printer that sends a graph to a graphic file. A format of Encapsulated PostScript (EPS) is proven to be successful. For details about how to install a virtual printer that sends a graph to a disk file, consult the manual of the Windows operation system.

5.9. Saving output files

5.9.1. Saving whole Mesh files

Purpose: To save a mesh file that can be used as input to the TOUGH2 simulator.

How: 1. Click File menu bar, and Point to Save, and then select 3-D Mesh;

2. Select a file from the mesh file list (if you want to overwrite an existing mesh file) or type a new file name into the file name box;

3. Click Ok button.

Tips: Use "mesh" as the extension of the mesh file.

5.9.2. Save a subset of a 3-D mesh and the selection information

Purpose: To save a subgrid selected by selecting columns method into a mesh file that can be used as input to TOUGH2 simulator, and to save the configuration of the selected subgrid (columns and segments) if a user requires so.

How: 1. Click File in the menu bar, and point to Save and then SelectedMesh;

2. Click ForTOUGH2;

3. Select a file from the mesh file list (if you want to overwrite an existing mesh file) or type a new file name into the file name box;

4. Click Ok button;

5. Click the Yes button if you want to save the configuration of the selected subgrid, and choose a proper file. Otherwise, click the No button.

Tips: 1. If no sub grid is selected, clicking Save SelectedMesh will cause WinGridder to warn you that no column is selected. You can either select a subgrid interactively or load the selection information from an existing previously saved disk file.

2. Use "scc" as the extension of the file that contains configuration information of a selected sub grid. Always save the configuration file if you are not sure if it has been saved before so that you do not have to select the sub grid interactively again.

5.9.3. Save information of a subset of a 3-D mesh for quality checking and selection information

Purpose: To save a subgrid selected by selecting columns into a disk file that can be used to check how well the three-dimensional grid represents the geological model of a site, and to save the configuration of the selected subgrid (columns and segments) if a user so requires.

How:
1. Click File menu bar and point to Save and then SelectedMesh;
2. Click ForChecking;
3. Select a file from the checking file list (if you want to overwrite an existing mesh file) or type a new file name (extension "mck") into the file name box;
4. Click Ok button;
5. Click Yes if you want to save the configuration of the selected subgrid and choose a proper file. Otherwise, click No.

Tips:
1. If no sub-grid is selected, clicking Save SelectedMesh will cause WinGridder to warn you that no column is selected. You can either select a subgrid interactively or load the selection information from an existing previously saved disk file.
2. Use "scc" as the extension of the file containing the configuration information of a selected subgrid.

5.9.4. Saving checking file of Boreholes

Purpose: To save a checking file of all cells in the borehole columns used to compare with the geologic data of those boreholes.

How:
1. Click File menu bar and select Save and then select Mesh of Boreholes;
2. Select a file from the checking file list (if you want to overwrite an existing checking file) or type a new file name (extension "mck") into the file name box;
3. Click Ok button.

Tips: A text file called "Borehole_Column.txt" will be saved as a byproduct, which contains a list of the LBNL names of the columns and the corresponding borehole names. A column is referred as a borehole column when its "kind" property contains the keyword "Bore".

5.9.5. Saving a List of Fault columns

Purpose: To save a list of LBNL names for the fault columns and the corresponding fault names.

How: 1. Click File in the menu bar, point to Save, and then select "List of Fault Columns";

2. Select a file from the file list (if you want to overwrite an existing listing file) or type a new file name (extension "lst") into the file name box;

3. Click Ok button.

Tips: WinGridder selects all columns whose "kind" property contains a keyword "Fault" as fault columns.

5.9.6. Saving a List of borehole columns

Purpose: To save a list of LBNL names for the fault columns and the corresponding borehole names.

How: 1. Click File in the menu bar, point to Save, and then select List of Borehole Columns;

2. Select a file from the file list (if you want to overwrite an existing listing file) or type a new file name (extension "lst") into the file name box;

3. Click Ok button.

4. If borehole data do not exist, WinGridder will warn you and let you explore your hard disk to load a proper file of borehole data.

Tips: In contrast to the "Borehole_Column.txt", WinGridder will in this case find the nearest column for each borehole regardless of the column's "kind" property and save the list to the listing file.

5.9.7. Saving a List of columns west of Solitario Canyon Fault

Purpose: To save a list of names of the columns that are west of Solitario Canyon Fault. This fault is one of the major faults at the Yucca Mountain site, transecting the simulation domain from north to south.

How: 1. Click File in the menu bar, point to Save, and then click on "List of columns west of Sol. Canyon";

2. Click the fault file that defines the trace of Solitario Canyon Fault;

3. Click Ok button;

4. Click the file from the file list (if you want to overwrite an existing file) or type a new file name (extension "lst") into the file name box;

5. Click Ok button.

Tips: WinGridder first builds a polygon by combining the fault data and the west boundary of the simulation domain. It then checks if a column is within the polygon to decide whether it is on west of the fault. Therefore, it is possible to pick columns west of a fault other than Solitario Canyon Fault by selecting another proper fault data file. Furthermore, you may use a dummy fault to serve this purpose. However, interactively selecting columns (an area) may be more flexible if you want to pick a subgrid.

5.9.8. Saving a List of repository cells

Purpose: To list all cells whose "kind" property has a keyword "InRepo" in a disk file. For details, see "List of Repository Cells" in Section 4.2.2.

How: 1. Click File in the menu bar, point to Save, and then click "Repository Cell";

2. Click the file from the file list (if you want to overwrite an existing file) or type a new file name (extension "lst") into the file name box;

3. Click Ok button.

Tips: This list of repository cells does not include the cells that are fault-related (i.e., fault cells or side cells) even though they are located in the repository drift horizon.

5.9.9. Saving the list of cells of a particular layer (including repository horizon)

Purpose: To save the information of all cells that belong to a user-specified layer or repository horizon. For details about the format of this file, see List of Cells for a Specified Layer (including repository horizon) in Section 4.2.2.

How: 1. Click the File menu bar, point to Save, and then click Pick Cells by layer;

2. Click the file from the file list (if you want to overwrite an existing file) or type a new file name (extension "lst") into the file name box;

3. Click Ok button.

Tips: The number of cells in this file always equals the number of columns because each column will contribute a cell, no matter whether the specified layer is missing or not at a particular column.

5.9.10. Saving a Segment file in a format compatible with Amesh

Purpose: To save the updated information of segments in a format compatible with Amesh that can be used by previous postprocessing programs. As a result of modifications of segments in design of a column scheme, the information in the segment file saved by Amesh is no longer correct.

How:
1. Click the File menu bar, point to Save, and then click "Segments as amesh";
2. Click the file from the list (if you want to overwrite an existing file) or type a new file name (extension "grd") into the file name box;
3. Click Ok button.

Tips: A better way to output a 2-D grid graph is to print the graph to a file.

5.9.11. Saving mesh files for Dual-permeability grid Generator

Purpose: To save mesh files readable by the dual-permeability grid generator.

How:
1. Click the File menu bar, point to Save, and then click "Mesh for DualK Generator".
   The WinGridder will ask you to enter the filename of the "m2k" file.

Tips: If you want use the dual-permeability grid generator embedded in the WinGridder, you do not have to use this function first. Instead, you can directly use the generator (see Section 5.4.4); the WinGridder will save the files there.

5.9.12. Saving mesh files for DCPT

Purpose: To save mesh files readable by the DCPT (Dual Continua Particle Tracker) V1.0.

How: Click the File bar, point to Save, and then click "Grid for Particle Tracking". The WinGridder will ask you to enter a filename.

Tips: This function is designed for DCPT users only.

5.10. Tools

5.10.1. Text Editor

Purpose: To use a text editor to view, edit, and print a text file that may be an input, output, or intermediate file.

How: Click Tools menu bar; Click TextEditor

A Notepad or WinWord will appear on the screen.

Tips: WinGridder actually calls Windows' Notepad or WinWord if the size of the file is too large for Notepad. Because the text editor is a third-party component, WinGridder will not track its status after calling it. It is your responsibility to close it after finishing your job.

5.10.2. Tecplot

Purpose: To start Tecplot.

How: 1. Click Tools menu bar;

2. Click Tecplot. Tecplot will appear on the screen.

Tips: Tecplot is a third part component. You have to install it on your PC before you can use it through WinGridder. It is the user's responsibility to close it after finishing your job.

5.10.3. Quality checking of 3-D cells

Purpose: To check vertical connections of 3-D cells and lateral connections of repository cells. WinGridder will warn you if ether of following situations exists:

(1) Any cell has no cell below it except the bottom boundary cell in any column;

(2) Any cell has no bottom face (connection) except the bottom boundary cell in any column;

(3) Any repository cell only has one or fewer lateral connections.

How: 1. Click Tools on the menu bar;

2. Click Q-Checking;

3. Click 3DCells.

Tips: Do not use this function before the entire grid has been completely built.

5.10.4. Quality checking of 3-D connections

Purpose: To check if any duplicated lateral connections exist.

How: 1. Click Tools menu bar;

2. Click Q-Checking;

3. Click 3DCons.

Tips: Do not use this function before the entire grid has been completely built.

5.10.5. Quality checking of overlapped columns

Purpose: To check if any duplicated columns exist.

How: 1. Click Tools menu bar;

2. Click Q-Checking;

3. Click Overlap Column.

Tips: This is a slow process. Use it only when necessary.

6. Examples

6.1. Site scale 3-D grids of Yucca Mountain

Site-scale 3-D grids of Yucca Mountain are built for UZ modeling efforts for the Yucca Mountain Project at LBNL. The UZ Model represents known geological information about Yucca Mountain. In addition to lithostratigraphic and structural data, relevant hydrogeological information pertaining to the distribution of mineral alteration below the repository horizon is essential to any comprehensive flow and transport model. The potential of clays and zeolites to greatly reduce permeability has important implications for fluid flow-paths and travel times, and for perched water development. Moreover, the sorption potential of zeolites leads to retardation of nuclides migrating away from the repository horizon. A new site scale 3-D grid is needed to incorporate geologic and mineral information into the UZ flow and transport model. Three new requirements of the grid are realizations of the inclined faults, the mineral alteration zones, and local denser grid within repository horizon.

6.1.1. 2-D Grid (column scheme) Development

The 2-D (plan-view) grid consists of nodal point locations and line segments defining the location and shape of the 3-D gridblock columns. The 2-D grid is generated with WinGridder1.0 software which uses Delaunay triangulation techniques. Grid generation is an iterative process of moving, deleting, and adding nodal points to achieve a desired array.

All relevant files should be in the same working directory. The relevant input files for 2-D grid generation of this example include:

File Name	Description
DomainBound.bnd	UZ model boundary coordinates
RepoThin.bnd	repository boundary coordinates
BorAlcNichTie.hol	borehole, alcove, niche coordinates
ECRB.dft	ECRB map-view coordinates
ESF.dft	ESF map-view coordinates
FLTsolcan.fut	Solitario Canyon Fault coordinates
FLTsolwest.fut	Solitario Canyon Fault - West coordinates
FLTsoljfat.fut	coordinates of fault joining Solitario Canyon and Fatigue Wash faults
FLTsever.fut	Sever Wash Fault coordinates
FLTpagany.fut	Pagany Wash Fault coordinates
FLTdrill.fut	Drill Hole Wash Fault coordinates
FLTdune.fut	Dune Wash Fault coordinates
FLTdunew1.fut	Dune - West1 fault coordinates
FLTdunex.fut	Dune - X fault coordinates
FLTghost.fut	Ghost Dance Fault coordinates
FLTghostw.fut	Ghost - West fault coordinates
FLTimb.fut	"Imbricate" Fault coordinates
FLTsplays.fut	Splay "S" fault coordinates
FLTsundance.fut	Sundance Fault coordinates
FLTsplayn.fut	Splay "N" fault coordinates
FLTsplayg.fut	Splay "G" fault coordinates

The following steps were taken to produce the plan-view UZ Model grid:

Step 1: Load lateral boundary information: File → Load → Bound

Select files "DomainBound.bnd" and "RepoThin99.bnd." These files define the UZ Model domain and repository outline, respectively.

Step 2: Load fault information: File → Load → Faults

Select all faults within the UZ Model domain, but not those coincidental with model boundaries (*.fut).

Step 3: Load borehole information: File → Load → Borehole

Select the file "BorAlcNichTie.hol"

Step 4: Load ECRB and ESF information: File → Load → Drifts

Select files "ECRB.dft" and "ESF.dft"

Step 5: From the menu bar, select: Design → 2D

Under "Select an Object," choose Domainbound.

Enter an "Average Distance" (node spacing) of 300 (meters).

Pop-up windows appear:

Rectangular Mesh? No

Random? Yes

Under "Select an Object," choose Repository.

Enter an "Average Distance" of 200 (meters).

Pop-up windows appear:

Rectangular mesh?: Yes

Enter Y to X ratio: 1

Under "Select an Object," choose ECRB3Row (this will add nodes along the ECRB similar to the way faults are gridded - i.e., a 3-row approach).

Enter an "Average Distance" of 100 (meters).

Enter Y to X ratio: 1

Rectangular mesh?: Yes

Under "Select an Object," choose ESF3Row (this will add nodes along the ESF similar to the way faults are gridded - i.e., a 3-row approach).

Enter an "Average Distance" of 100 (meters).

Enter Y to X ratio: 1

Rectangular mesh?: Yes

Under "Select an Object," choose All Faults.

Enter an "Average Distance" of 100 (meters).

Pop-up window appears:

Enter a fault width: 30 (meters).

Under "Select an Object," choose Boreholes.

Enter an "Average Distance" of 30 (meters).

Step 6: At this point, the location of certain nodal points created for each object (e.g., domain, repository, faults, and boreholes) are modified (e.g., deleted, moved, or added) to achieve a desired nodal point array. The nodal array is saved as: "UZ99_1.in"

Step 7: From the menu bar, select: Generate → 2D

Load the file "UZ99_1.in." The program runs and creates the file "UZ99_1.grd."

Step 8: Load the grid file "UZ99_1.grd" and remove any incorrect segments from the 2-D grid by selecting: Design → Remove Bad Segments

This targets fault segments, in particular; however, the user may also define a "minimum ratio" for a segment, so that any values occurring in the grid below this ratio will also be deleted from the grid. In this case, a minimum ratio of 5% was specified.

Step 9: Save the 2-D grid: Project→ Save → 2D

The project is saved as "UZ99_2.prj"

6.1.2. 3-D Grid Development

The relevant files for 3-D grid generation include:

File Name	Description
UZ99_2.prj	the 2-D grid project
DomainBound.bnd	UZ Model domain boundary
RepoThin99.bnd	repository boundary and elevation information
DualMaxH.bnd	boundary within which additional vertical resolution will be added
LayerList.txt	list of all elevation & thickness files needed for the 3-D grid
Layerorder.dat	compilation of all header information contained in the LayerList files
UZ99.roc	assigns a rock material type to each layer file

The following steps were taken to produce the 3-d grid:

Step 1: Open WinGridder1.0, Load lateral boundary information: Select files "DomainBound.bnd", "DualMaxH.bnd", and "RepoThin99.bnd" . and then load the 2-D grid project: "UZ99_2.prj"

Step 2: From the menu bar, select: Design → Assign DualMaxH Columns

Select "DualMaxHRegion"

Input critical elevation below which the MaxH will be different: 1,111 masl

(This just marks the columns for which additional vertical resolution will be added.)

Step 3: From the menu bar, select Edit → Update Repo Columns

Step 4: From the menu bar, select: Generate → 3D → Cells/Vertical Cons Only

A window pops up and asks the user if they want to automatically save the project. For large projects (such as this one), select "Yes." The project is named: "UZ99_2_3D.prj"

Pop-up windows appear for which the following information is specified:

Use long LBNL name (8 characters)? If no, default of 5 characters is used: select "No"

Minimum thickness of a cell: 1.5 (meters)

Maximum thickness of a cell: 60 (meters)

Maximum thickness below the repository: 20 (meters)

Do you want local refinement in the repository area? No

Load geologic layer file. File → Load → "LayerList" then "UZ99.roc"

Do boreholes have their own layering information? Select "No".

Do you want to reset the water table location west of Solitario Canyon Fault? Select "Yes," and type in the elevation difference (west-east): 46 (meters).

Also, select/open the fault file: "FLTsolcan.fut"

Step 5: From the menu bar select: Generate → 3D → Lateral Cons Only

A window pops up and asks the user if they want to laterally connect top cells and laterally connect bottom cells even if these cells belong to different layers? Select "No"

Specify a minimum cell thickness: 1.5 (meters)

The program runs and saves the 3-D grid as: "UZ99_2_3D.prj"

6.1.3. Outputs

From the menu bar, select File→Save→3D mesh. The output TOUGH2 file created is called: "UZ99_2_3D.mesh".

A map view of the generated 3-D grid is shown in Figure 6.1.3-1.

6.1.4 Validation of the Grid

Consistency between the generated 3-D grid and the geologic and hydrogeologic input data is checked by comparing layer elevations and thicknesses through 1-D and 2-D visual inspections. Figures 6.1.4-1 and 6.1.4-2 show examples of these comparisons. The input data come from the Geologic Framework Model, GFM3.1, of Yucca Mountain. Numerical grid information generated by WinGridder V1.0 is for the 3-D, site-scale, unsaturated-zone model of Yucca Mountain.

UZ Model layer contact elevations and thicknesses are determined based on the value of the closest GFM3.1 data point (GFM3.1 data have a regular spacing of 61 × 61 m). In other words, no interpolation of GFM3.1 data to the exact location of a UZ Model node is performed. The estimated maximum error in layer contact elevations at column centers associated with this “nearest-value” approach is about 5 m, assuming that the geologic layers dip 10 degrees. This amount of potential error is considered insignificant to grid development and subsequent site-scale model simulation activities because lateral column dimensions almost always exceed 61 × 61 m, thus, encompassing the nearest GFM3.1 data point.

For 1-D vertical borehole comparisons between UZ Model grid columns and GFM3.1 borehole lithostratigraphy, a grid validation criterion of plus-or-minus 5 m is established, based on the expected maximum error discussed in the paragraph above. Figure 6.1.4-1 shows a comparison between GFM3.1 and UZ Model layering at eleven different vertical borehole locations. Columns of gridcells corresponding to the location of the surface-based vertical boreholes were extracted from the UZ Model grid, and the elevation of the grid-layer contacts within these columns were subtracted from the observed borehole lithostratigraphic contacts from GFM3.1. This comparison shows that the layer elevations from the UZ Model grid closely match the GFM3.1 input data. The accuracy of the 3-D UZ Model grid is within 5 m (typically within 3 m) for these borehole locations. Line discontinuities indicate missing or pinched-out layers.

The second type of grid validation involves visual, cross-sectional comparisons between GFM3.1 and columns of gridcells from the UZ Model taken along the same cross-sectional traverse to ensure that gridcell material names and elevations are correct. One of the goals of this activity is to evaluate the UZ Model's representation of stratigraphy immediately adjacent to fault planes—an area in which numerical gridding can become extremely complex. Figure 6.1.4-2 shows one of these cross sections.

Earthvision V4.0 (STN: 30035 V4.0) is used to extract cross-sectional data (layer contact elevations and fault locations) from GFM3.1 surface/horizon grids. These GFM3.1 data are used to develop the base map in the cross-sectional figure. To simplify the comparison, certain GFM3.1 layers are combined into single layers if they were grouped together in the UZ Model. Overlain on the base map are columns depicting the UZ Model representation of these layers along the same cross section. Data defining the top and bottom elevations of each hydrogeologic layer in the UZ Model were obtained as output directly from WinGridder V1.0. For simplicity, the actual grid resolution *within* hydrogeologic layers is not shown in the comparison plot. UZ Model gridcells are color coded to match the corresponding GFM3.1 layer. Note that the column width shown in the comparison plots is not to scale and that all grid layer interfaces in the UZ Model are actually horizontal, rather than tilted as shown along faults.

Figure 6.1.4-2 is an east-west cross section that intersects Solitario Canyon, Drill Hole Wash, and Pagany Wash faults. Overall, the layer thickness and elevation match is good between the UZ Model grid and GFM3.1. Vertical displacement along the Solitario Canyon fault appears to be about 20 m less in the UZ Model than is seen in GFM3.1 along this cross section; however, this is only an artifact of the plotting technique. Recall that the column width shown in Figure 6.1.4-2 is not to scale. Gridcells adjacent to nonvertical fault zones may be a few hundreds of meters wide (as is the case along the Solitario Canyon fault in this cross section). Thus, the elevations of the gridcells shown in Figure 6.1.4-2 are actually average elevations over larger lateral distances than suggested by the column width shown. With this in mind, it can be concluded that the elevation of UZ Model gridcells are in reasonable agreement with GFM3.1 data along this cross section. The only other apparent discrepancy is in the UZ-14 grid column and its adjacent columns. In this location, the UZ Model imposes a thickness for geologic layer Tptpln (corresponding to model layers "tsw36" and "tsw37") that is approximately twice the thickness from GFM3.1. Thus, the layer thickness difference observed in the comparison is not an error in the gridding process.

A third method of grid validation involves an analysis of vertical and lateral connection information between gridcells. This independent evaluation of the connections within the 3-D grid is performed using the TOUGH2 code and a post processor (i.e., EXT V1.1). TOUGH2 output from a simple simulation (e.g., one using either a fully saturated or completely dry condition) is processed using the ext V1.1, and the resulting saturation contours are displayed along the cross section being checked. If gridcell connections are missing from the UZ Model grid along this cross section, a "hole" is shown in the saturation contour plot resulting from the absence of data for that area of the grid. No "holes" were observed in the cross sections examined.

6.2 A simple 3-D grid

This grid is designed to test the accuracy of geometrical calculations of WinGridder. The domain is 30 × 30 × 100 meters. There are two layers of rock. Each of them has a thickness of 50 meters. Various geometrical features of the generated grid can be validated against theoretical values for this simple 3-D grid. The features include the volumes of cells, the interface areas of connections, the distance of a node to the interface, the layering interface represented by the grid, and the cosine of the angle between a connection and the vertical direction (cosZ). The 3-D grid should consist of 9 columns. Each column has 10 cells (excluding the boundary cells that are purely for the purpose of assigning the boundary conditions). As results, the volume of each cell should have a size of 10 × 10 × 10 meters (=1000 m^3); the interface area of each connection should be 10 × 10 (=100 m^2); the distance of a node to the interface should be 5 meters; the layering interface should be at the elevation of 50 meters; and the absolute values of cosZ should be 1.0 (or –1.0) for vertical connections and 0 for horizontal connections, respectively.

6.2.1 Development of the grid

The following steps were used to generate this grid:

Step 1. Generate the 2-D grid:

Load lateral boundary information: File→Load→ Bound

Select file "Sample3D.bnd". This file defines the lateral boundary of the domain.

From menu bar, select Design 2D

Under "Select an Object," choose DomainBound; Specify an Average Distance (node spacing): 10 (distance in meters) and click Add Nodes. A number of windows pop up. Enter the following information:

Enter angle at which to align nodes (positive to X axis): 0

Enter Y to X ratio: 1

Rectangular mesh?: Yes

Click Save and the file is saved as "Sample3D.in"

From the menu bar, select: Generate 2D

Step 2. Load the 2-D grid information:

From the menu bar, select File→Load→Grid

Select file "Sample3D.grd"

Step 3. Generate the 3-D grid:

From the menu bar, select: Generate→ 3D→Full mesh

A window pops up and asks the user if they want to automatically save the project, select "Yes."

Enter "Sample3D.prj" as the project name and then click Save button.

A number of windows appear for which the user specifies the following:

Use long LBNL name (8 characters)?: select "No"

Minimum thickness of a cell: 1 (meter)

Maximum thickness of a cell: 10 (meters)

Do you want local refinement in the repository area? Select No

Load geological layer file: Click OK, and select "LayerList.txt" then "Layers.roc"

Do you want to reset the water table location west of Solitario Canyon Fault? Select "No"

Do boreholes have own layer configuration? Select "No"

Do you want to laterally connect top cells and laterally connect bottom cell even if these cells belong to different layers? Select "No"

6.2.2 Output

From the menu bar, select File→Save→3D mesh. The mesh file is saved as "Sample3D.mesh."

6.2.3 Validation of the Grid

Comparing the WinGridder output to the hand-calculated values in Section 6.2 checks consistency between the generated 3-D grid and the theoretical geometry data of the grid. See Figure 6.2.3-1 for a sample output from WinGridder based on the simple grid in Section 6.2. Exact matches are found for the volumes of cells (highlighted column under ELEME, 1000 m^3), the interface areas of connections (second highlighted column under CONNE, 100 m^2), the distance of a node to the interface (first highlighted column and first boxed column under CONNE, 5 m), the layering interface represented by the grid (two boxed rows under ELEME, height is midpoint between 45 m elevation and 55 m, so 50 m), and cosZ (second boxed column under CONNE, -1.0 for vertical connections and 0.0 for horizontal connections).

6.3 A dual-permeability grid

This dual-permeability grid is for verifying the accuracy of the dual-permeability grid generator as a component of the WinGridder. To be able to check the quality by hand calculating, a simple and regular grid is used.

6.3.1 Development of a dual-permeability grid

The involved files are:

Input files:

'framtr.fp'	the fracture property file
'2kgrid.m2k'	the ECM mesh file (cells and connections)
'2kgrid.idx'	the ECM mesh file (connection indexes)

Output files:

'2kgrid.2ko'	the output file for cross reference of the cell names in two meshes

'2kgrid.2ke' the dual-K mesh file (cells)
'2kgrid.2kc' the dual-K mesh file (connections)

Steps:

Select Generate →3D→Dual-K Mesh, the WinGridder will ask you to select a fracture property file which you should select “framtr.dat”;

Click Yes for question "Has ECM mesh been saved for 2K grid?" box, then select the file “2kgrid.m2k” in file selection box, and the WinGridder will automatically load “2kgrid.idx” after loading “2krid.m2k”.

The output files “2kgrid.2ke” and “2kgrid.2kc” contain the cells and the connections, respectively, in the format readable by TOUGH family codes.

6.3.2 Validation of the dual-permeability grid

The original ECM grid is a 3 × 3 × 3 mesh (a total of 27 cells) with 30 × 30 × 30 m domain. The fracture volume is 1% of the bulk volume. The fracture spacing is 1 meter while the interface area is 1.98 m^2/m^3. Consequently, the resulting volumes of the matrix cells and the fracture cells should be 990 m^3 and 10 m^3, respectively (see equations 3.1 and 3.2). The distances to the interfaces are 0 and 0.1667 m for the fracture nodes and the matrix nodes, respectively. The interface areas should be 1,980 m^2 per cell. The results in Figure 6.3.2-1 (2kgrid.2ke) and Figure 6.3.2-2 (2kgrid.2kc) show that the dual-permeability grid generator correctly calculated the above parameters of the dual-permeability grid and meets all criteria.

7. Summary and Conclusions

WinGridder is designed for generating 3-D grids that conserve the physical relationships between objects with a finite number of cells. It can save mesh files that can be used as part of the input to the TOUGH family of codes. WinGridder can also output additional grid information for various purposes, such as grid inspection and visualization. It has user-friendly graphical interfaces. With its interactive editing functions, a user can easily check and modify a layout of columns.

The examples in Section 6 are by no means exhaustive. Users should take these examples as illustrations of how to use various functions provided by WinGridder to build grids.

The software has been extensively tested, and the generated grids have been validated during modeling efforts in the UZ modeling group at LBNL. However, because grid generation is a complex process, understanding of the conceptual geological model and grid generation procedure is needed to generate numerically valid grids.

Because WinGridder is in its first version, many important functions are still under development and as a result, bugs or program failure may occur when some of these functions are executed. Any comments are welcome.

References

Haukwa, Charles, 2000. AMESH: a mesh creating program for the integral finite difference method. Lawrence Berkeley National Laboratory Report LBNL-45284. Berkeley, California: Lawrence Berkeley National Laboratory.

Hinds, J. and L. Pan, 1999, FY 99 UZ Model calibration grid; AMR: U0000 Development of Numerical Grids for UZ Flow and Transport Modeling. Analysis/Model Report ANL-NBS-HS-000015 REV00B, CRWMS M&O, Las Vegas, Nevada.

Pruess, K., 1991. TOUGH2 – A general purpose numerical simulator for multiphase fluid and heat flow, Lawrence Berkeley Laboratory Report LBL-29400. Berkeley, California: Lawrence Berkeley National Laboratory.

Pruess, K., 1987. TOUGH User's Guide, Lawrence Berkeley Laboratory Report LBL-20700. Berkeley, California: Lawrence Berkeley National Laboratory.

Pruess, K., 1983. GMINC – A mesh generator for flow simulations in fractured reservoirs, Lawrence Berkeley Laboratory Report LBL-15227. Berkeley, California: Lawrence Berkeley National Laboratory.

List of Figures