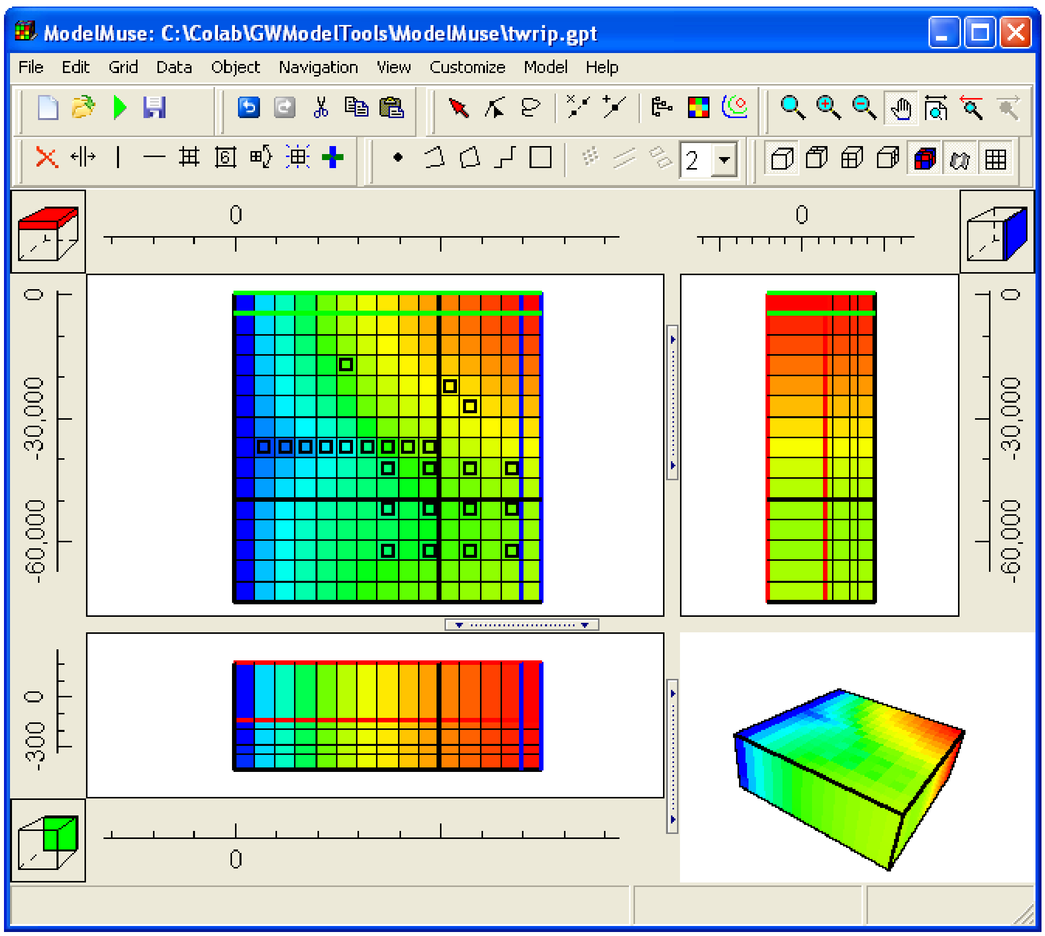
USGS
science for a changing world

ModelMuse—A Graphical User Interface for MODFLOW–2005 and PHAST

By Richard B. Winston

Chapter 29 *of*
Section A, Ground Water—Book 6, Modeling Techniques



Techniques and Methods 6–A29

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
KEN SALAZAR, Secretary

U.S. Geological Survey
Suzette M. Kimball, Acting Director

U.S. Geological Survey, Reston, Virginia: 2009

For product and ordering information:
World Wide Web: *http://www.usgs.gov/pubprod*
Telephone: 1-888-ASK-USGS

For more information on the USGS—the Federal source for science about the Earth,
its natural and living resources, natural hazards, and the environment:
World Wide Web: *http://www.usgs.gov*
Telephone: 1-888-ASK-USGS

Cover: View of the example model for MODFLOW–2005 described in the MODFLOW–2005 documentation. The model was imported into ModelMuse and the grid was colored with the calculated head.

Suggested citation:
Winston, R.B., 2009, ModelMuse—A graphical user interface for MODFLOW–2005 and PHAST: U.S. Geological Survey Techniques and Methods 6–A29, 52 p., available only online at http://pubs.usgs.gov/tm/tm6A29.

Contents

Figures

Abbreviations and Acronyms

m	meter
CPU	central processing unit
DXF	Drawing Exchange Format
GB	gigabyte
GHz	gigahertz
IFACE	auxiliary parameter to input boundary faces

MB	megabyte
MODFLOW	modular groundwater flow model
PHAST	**PH**REEQC **A**nd H**ST**3D computer program for simulating groundwater flow, solute transport, and multicomponent geochemical reactions

Preface

This report describes the U.S. Geological Survey graphical user interface for MODFLOW and PHAST (ModelMuse). The performance of the program has been tested in a variety of applications. Future applications, however, might reveal errors that were not detected in the test simulations. Users are requested to send notification of any errors found in this report or the model program to:

Office of Ground Water
U.S. Geological Survey
411 National Center
Reston, VA 20192
(703) 648-5001

The latest version of the model program and this report can be obtained using the Internet at address: *http://water.usgs.gov/software/*.

ModelMuse—A Graphical User Interface for MODFLOW–2005 and PHAST

By Richard B. Winston

Abstract

ModelMuse is a graphical user interface (GUI) for the U.S. Geological Survey (USGS) models MODFLOW–2005 and PHAST. This software package provides a GUI for creating the flow and transport input file for PHAST and the input files for MODFLOW–2005. In ModelMuse, the spatial data for the model is independent of the grid, and the temporal data is independent of the stress periods. Being able to input these data independently allows the user to redefine the spatial and temporal discretization at will. This report describes the basic concepts required to work with ModelMuse. These basic concepts include the model grid, data sets, formulas, objects, the method used to assign values to data sets, and model features.

The ModelMuse main window has a top, front, and side view of the model that can be used for editing the model, and a 3–D view of the model that can be used to display properties of the model. ModelMuse has tools to generate and edit the model grid. It also has a variety of interpolation methods and geographic functions that can be used to help define the spatial variability of the model. ModelMuse can be used to execute both MODFLOW–2005 and PHAST and can also display the results of MODFLOW–2005 models. An example of using ModelMuse with MODFLOW–2005 is included in this report. Several additional examples are described in the help system for ModelMuse, which can be accessed from the Help menu.

Introduction

ModelMuse is a graphical user interface (GUI) for MODFLOW–2005 (Harbaugh, 2005) and PHAST (Parkhurst and others, 2004). MODFLOW–2005 is a three-dimensional finite-difference groundwater model. It simulates steady and nonsteady flow in an irregularly shaped flow system in which aquifer layers can be confined, unconfined, or a combination of confined and unconfined. PHAST simulates multi-component, reactive solute transport in three-dimensional saturated groundwater flow systems.

ModelMuse is based on GoPhast (Winston, 2006). ModelMuse allows the user to define the spatial input for the models by drawing points, lines, or polygons on top, front, and side views of the model domain. These objects can have up to two associated formulas that define their extent perpendicular to the view plane, allowing the objects to be three-dimensional. Formulas are also used to specify the values of spatial data (data sets) both globally and for individual objects. Objects can be used to specify the values of data sets independent of the spatial and temporal discretization of the model. Thus, the grid and simulation periods for the model can be changed without respecifying spatial data pertaining to the hydrogeologic framework and boundary conditions. The points, lines, and polygons can assign data set properties at locations that are enclosed or intersected by them or by interpolation among objects using several interpolation algorithms. Data for the model can be imported from a variety of data sources and model results can be viewed in ModelMuse. This report describes the basic operation of ModelMuse along with an example model. Additional information and examples are provided in the ModelMuse help system, which can be accessed from the Help menu.

Once the model has been defined in ModelMuse, the user can create the input files for the model by selecting **File|Export** and then export either the MODFLOW input files or the PHAST transport input files. The user has the option to execute the model immediately once the input files are exported.

In cases where the input files for MODFLOW–2000 and MODFLOW–2005 are identical, it may be possible to use ModelMuse to create input files for MODFLOW–2000. However, ModelMuse has not been extensively tested with MODFLOW–2000. Some differences between MODFLOW–2000 and MODFLOW–2005 include the formats of the input files for the observation process and the absence of the Unsaturated Zone Flow (UZF) package in MODFLOW–2000.

The current version of ModelMuse does not support all the options in MODFLOW–2005. Additional options and other programs may be supported in future versions of ModelMuse.

ModelMuse stores all its data in a single file. Several file formats are supported. Of these, the most commonly used are text files with the extension ".gpt" and compressed binary files with the extension ".mmZLib".

In ancient Greece and Rome, the Muses were thought, by some, to provide the inspiration for music, poetry, and the arts. The composers, poets, and other artists, however, still had to do the hard work of turning that inspiration into an actual work of art. It would be great if ModelMuse could do the same for modelers—provide the key insight required to allow the system to be quickly and effectively modeled. ModelMuse can not do that; it is not smart enough. What it can do is take over some of the mundane parts of the modeling process and make them much easier and faster. By doing so, ModelMuse allows the modeler more time to think, to observe, to analyze, to experiment, and to generate the needed inspiration.

Quick Start Guide

When the user starts ModelMuse, the **Start-Up** dialog box will appear in which the user can choose to (1) create a new model, (2) import a model, or (3) open an existing model. If the user chooses to create a new model, the user can create a grid for the model in the **Initial Grid** dialog box. The user can also choose to skip creating the grid and create it later in a variety of different ways. The first time ModelMuse is started on a computer, a video will play. The video can be played again later by selecting **Help|Introductory Video**.

MODFLOW Models

With MODFLOW models, the user must choose which packages to use in the model in the **Model|MODFLOW Packages** dialog box. The packages define the types of boundary conditions that can appear in the model, the method used to solve the model equations, and various other aspects of the model.

Stress periods for the model are set up in the **Model|MODFLOW Time** dialog box. The stress periods define time periods in the model during which the stresses on the model are kept constant (with a few exceptions).

Layers in the model can be confined or convertible between confined and unconfined. This and other properties can be set in the **Model|Layer Groups** dialog box.

Objects and Formulas are used to define the spatial data such as the distribution of hydraulic conductivity and the location of boundary conditions. The objects can be drawn on the top, front, or side views of the model or imported from external sources such as Shapefiles. It is also possible to import a background image to help when designing the model.

1. To execute the model, select **File|Export|MODFLOW Input Files**.
2. To view the model results, select **File|Import|Model Results...**

PHAST Models

If chemical reactions are to be simulated, the types of reactions to be simulated are chosen in the **PHAST Chemistry Options** dialog box. The time periods to be simulated in the model are specified in the **PHAST Time Control** dialog box. The choices related to the solution method are specified in the **PHAST Solution Method** dialog box.

Objects and Formulas are used to define the spatial data such as the distribution of hydraulic conductivity and the location of boundary conditions. The objects can be drawn on the top, front, or side views of the model or imported from external sources such as Shapefiles. It is also possible to import a background image to help when designing the model.

To create the flow input file for the model, select **File|Export|PHAST Input File**. The chemistry input file must be created outside of ModelMuse. Once all the PHAST input files are created, the user can execute PHAST from the command line or with a batch file. ModelMuse creates a batch file that can be used to execute PHAST at the same time that it creates the flow input file.

Model Viewer can be used to view the model results.

MODFLOW and PHAST Models

ModelMuse has a comprehensive help system. The help system can be accessed from the Help menu in the main window. In addition, most dialog boxes have a help button that can be used to access help on that dialog box. In addition to documenting ModelMuse, the help system also has comprehensive documentation of MODFLOW.

New users will need to understand the basic concepts behind ModelMuse to use it effectively. After learning the basic concepts behind ModelMuse, the example models may be helpful for new users. One example is presented in this documentation. Additional examples are included in the help system.

ModelMuse is a complex program and can sometimes assign values in ways that the user does not expect. If an unexpected value is assigned to an element or cell, check “Troubleshooting Data Set Values” in the documentation or in the help system.

Basic Concepts

To work effectively with ModelMuse, several basic concepts must be mastered. This section provides an introduction to those concepts and tells where more information about them may be found in this document.

The Grid

Both MODFLOW and PHAST use finite-difference techniques for spatial and temporal discretization. A grid is required for spatial discretization, but the grids in MODFLOW and PHAST differ in where data are calculated, where data are specified, and how the grid is numbered.

In ModelMuse, the grid can be rotated at an angle to the global coordinate system. The coordinate system for the grid is aligned with the grid lines, but has the same origin as the global coordinate system. The coordinates of a point in the global coordinate system are referred to as X, Y, and Z; the coordinates of a point in the grid coordinate system are referred to as X′, Y′, and Z (XPrime, YPrime, and Z). There is no Z′ because the grid is never rotated away from the horizontal plane. Coordinate values at the cursor location in both the global and grid coordinate system are displayed on the status bar of the main window of ModelMuse. More information about the grids for PHAST and MODFLOW can be found in the PHAST documentation (Parkhurst and others, 2004) and the MODFLOW–2005 documentation (Harbaugh, 2005).

ModelMuse can be used to create the grid; rotate it; and add, move, or remove grid lines. A variety of grid functions can be used in formulas.

PHAST Grid

PHAST uses a point-distributed grid. The nodes in this grid are at the corners of elements (fig. 1). Each node is surrounded by a cell that includes parts of one to eight elements (fig. 1). Boundary and initial conditions are specified by node; aquifer properties are specified by element. PHAST uses a standard right-hand coordinate system with the 1, 1, 1 layer, row, column in the closest lower left corner of the grid (fig. 3).

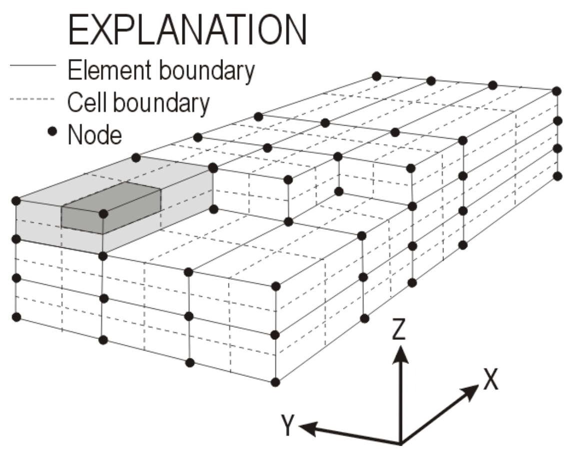


Figure 1. The grid in PHAST including nodes (black dots) and a light gray element and a dark gray cell. Solid lines represent element boundaries. Dashed lines represent cell boundaries.

MODFLOW Grid

The grid in MODFLOW uses block-centered nodes; the locations at which calculations are made are at the centers of blocks. Unlike PHAST, in MODFLOW the layers are not required to be flat. Instead, the top and bottom of each cell can be different (fig. 2). In MODFLOW the grid is numbered with 1, 1, 1 in the furthest upper left corner (fig. 3).

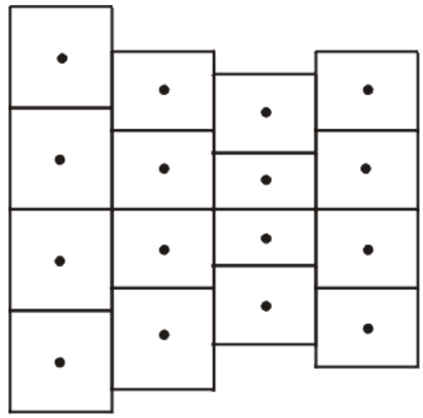


Figure 2. Side view of a MODFLOW grid showing non-uniform layer boundaries.

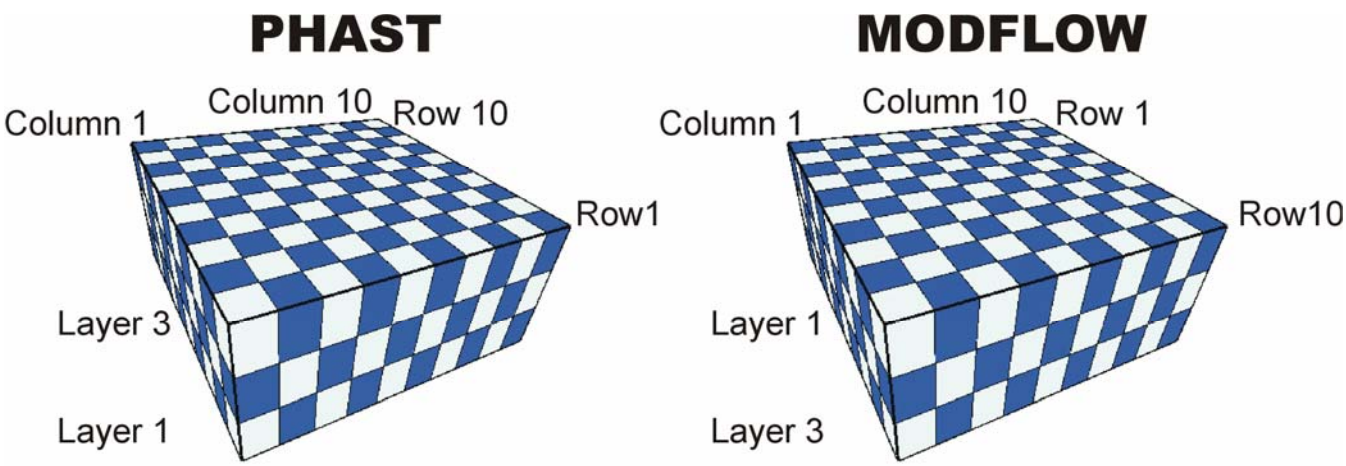


Figure 3. Difference in grid numbering between PHAST and MODFLOW.

In ModelMuse, groups of layers can be defined in the MODFLOW Layer Groups dialog box. Each group of layers shares a variety of common properties. If the model is to be a quasi-3–D model, some layer groups can be designated as nonsimulated in the MODFLOW Layer Groups dialog box. Data sets (described in the next section) are used to define the bottom of each layer group. An additional data set is used to define the top of the model.

Data Sets

Data sets are used in ModelMuse to represent spatially distributed data for each cell in MODFLOW or for each node or element in PHAST. Each data set represents a two-dimensional (2–D) or three-dimensional (3–D) array of values. 3–D data sets are defined for the entire extent of the model domain. 2–D data sets are defined for a top, front, or side projection of the model domain. If the number of rows, columns, or layers in the grid changes, the sizes of the data sets are also changed.

In addition to the data sets required by MODFLOW or PHAST, the user can create additional data sets. The user-defined data sets can be used in “Formulas” to define the distribution of values in the data sets that are required by MODFLOW or PHAST. See “Data Sets Dialog Box” in the help system for more information on data sets.

Data Sets in PHAST

Because PHAST requires that some data be assigned to elements and other data be assigned to nodes, some data sets used in PHAST will have a value for each element and others will have a value for each node. The initial water table in PHAST, which is a 2–D data set, is only defined for one layer of nodes in the model. The remaining data sets required by PHAST are 3–D.

Data Sets in MODFLOW

2–D data sets for the front and side views can not be used with MODFLOW. Data sets used in MODFLOW will always have a value for each cell. Geometrically, a MODFLOW cell is equivalent to a PHAST element.

For data sets in MODFLOW models for which parameters have been defined, the values of the data set will be determined by the parameters rather than the formula for the object. For such data sets, the phrase “(defined by parameters)” will appear after the name of the data set and the user will not be able to assign values for the data set directly. If the grid is colored with such a data set, ModelMuse will generate a formula that will mimic how the data set values are assigned by MODFLOW.

Formulas

Formulas are used to help define the distribution of values in data sets. One simple example of a formula would be just the name of another data set. For example, a valid formula for the Ky data set (which defines the hydraulic conductivity in the Y direction) would be “Kx.” (Kx is the data set that defines the hydraulic conductivity in the X direction.) Setting the formula for Ky to “Kx” would mean that the value of Ky in a given element would be equal to the value of Kx within that element.

Another simple example of a formula would be to set the formula for the Kz data set (Kz defines the hydraulic conductivity in the Z direction) to “Kx/10.” This formula would mean that in a given element, the value of Kz would be equal to the value of Kx in that element divided by 10.

Each data set has a “default formula” that is used to assign a value to each cell, node, or element when such values are not defined in some other way.

These examples only hint at the power of formulas. In addition to simple arithmetic operations, it is possible to use mathematical functions such as “sin” and “ln.” Geographic Information System (GIS) functions, logic functions, and functions related to the grid and objects are also available in formulas.

See “Formulas”, “Functions”, and “Formula Editor Dialog Box” in the help system for more information about formulas.

Formulas and MODFLOW Parameters.—In certain MODFLOW packages, such as the Layer Property Flow package, it is possible to define “parameters” that are used in combination with “multiplier arrays” and “zone arrays” to define the spatial variability of some input data such as the hydraulic conductivity. When parameters are defined, ModelMuse will not allow the user to define a formula for the related data set. Instead, it will generate a formula that reproduces what MODFLOW will do in assigning values to the input data.

Objects

Objects are collections of points, polylines (a series of connected line segments), and polygons drawn in the main window of ModelMuse or imported from external files. Objects can have one or more sections; each section is a point, polyline, or polygon. For example, a torus would be represented by an object with two concentric polygon sections. In the direction perpendicular to the plane in which it is drawn, an object can have formulas for zero, one or two surfaces. An object with zero surfaces is two-dimensional because it has only two coordinate directions defined. All objects, including point objects, with at least one surface defined are three-dimensional because they have three coordinate directions defined. Objects with two surfaces have an upper and lower surface making them three-dimensional. For example, a polygon with an upper and lower surface is a solid. The surfaces associated with an object need not be flat. The surfaces are defined by formulas that allow them to have virtually any shape. There is one limitation: none of the line segments defining an object can cross another line segment of the same section of the same object.

Objects drawn on the top view of the model that have two surfaces can apply to more than one layer at any one column-row location, whereas Objects drawn on the top view of the model that have one surface can apply to only a single layer at any one column-row location although the layer may vary among locations. Objects drawn on the front and side view behave analogously.

Objects are used to modify the default values of data sets and to set boundary conditions. Objects can be used to set values of data sets in any of three ways: (1) in two-dimensional data sets, values can be interpolated among objects (see Interpolation Methods); (2) values can be set for elements or cells whose centers or nodes are enclosed inside the object; and (3) values can be set for elements or cells intersected by the object. For the latter two methods, the order of the objects is important. Because each object overwrites previous values, only the last value applied takes effect.

See "Creating, Selecting, and Editing Objects in ModelMuse" and "Object Properties Dialog Box" in the help system for more information about Objects.

Assigning Values to Data Sets

ModelMuse assigns values to data sets using slightly different methods, depending on whether the model is a MODFLOW or PHAST model. However, in both cases formulas and objects are used to assign the values. Because values for data sets are specified using formulas and objects, the data for a given model are independent of the spatial discretization of the model. Values of a data set are needed when exporting the model input or when coloring the grid with the data set. Values for data sets are recalculated when the data are needed and the values for the data set are out of date. These values become out of date when any of the following occur:

1. The grid changes.
2. Any of the formulas used to set values of the data set change.
3. The interpolation method for the data set changes.
4. Any of the objects used to set the value of the data set are edited.
5. Any of the data sets on which the data set in question depends become out of date.

Assigning Values to Data Sets in PHAST

ModelMuse assigns values to data sets at nodes or elements in PHAST models using the following procedure.

1. First, a default value is assigned to every node or element by using either PHAST-style interpolation or mixtures (see "PHAST–Style Interpolation"), the selected interpolation method (see "Interpolation Methods"), or the default formula for the data set (see "Formulas" and "Data Sets Dialog Box" in the help system).
2. Next, each object that affects the data set is processed, and nodes or elements that are intersected or enclosed by each object are assigned values on the basis of PHAST-style interpolation or mixtures or by using the object's formula for the data set. Each object replaces values assigned previously by the default formula or by a previous object.

In PHAST, interpolation can be useful to specify the boundaries between geologic units (see example in fig. 4). To do this, the user first creates a data set for each interface between adjacent geologic units. Then the user creates point objects to specify the elevations of the interfaces at known locations. Interpolation can then be used to specify the elevations throughout the grid. These data sets for the elevations can then be used in the formulas for the upper and lower surfaces of "3D Objects" that define properties of aquifers.

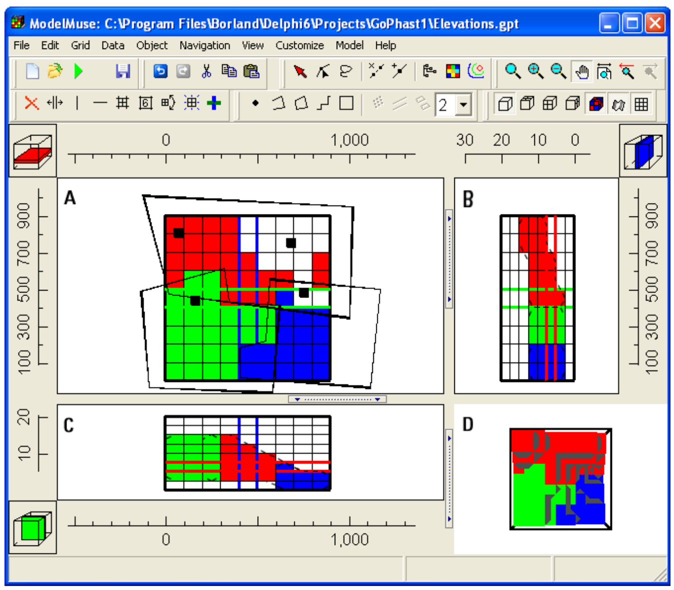


Figure 4. Example of 2–D data sets used to define the top and bottom of a geologic unit in PHAST—(A) Top view, (B) Side view, (C) Front view, (D) Three-dimensional view. In the top view (A), point objects (black squares) were used to specify the top and bottom of a geologic unit by interpolation. Polygons were then used to define the value of the hydraulic conductivity of that unit. The colored cells represent the different values of hydraulic conductivity. Note the sloping surfaces of the geologic unit visible in the front (C) and side (B) views of the model.

Assigning Values to Data Sets in MODFLOW

ModelMuse assigns values to data sets at cells in MODFLOW models using the following procedure.

1. First, a default value is assigned to every node or element by using either the selected interpolation method (see "Interpolation Methods"), or the default formula for the data set (see "Formulas" and "Data Sets Dialog Box" in the help system).

2. Next, each object that affects the data set is processed, and nodes or elements that are intersected or enclosed by each object are assigned values by using the object's formula for the data set. Each object replaces values assigned previously by the default formula or by a previous object.

In MODFLOW, 2–D data sets are typically used to define the upper and lower surfaces of grid layers rather than for defining objects that cross layer boundaries.

Model Features

In ModelMuse, "Model Features" are data that are only defined at certain locations. Most of them also vary with time. In most cases, Model Features are used to define the boundary conditions of the model. Model Features are treated similarly to Data Sets except that there are no default formulas for Model Features. Model Features are specified only with objects (points, lines, and polygons).

For both PHAST and MODFLOW, the user can specify different times for different Model Features and the times need not correspond to the boundaries of any previously defined stress period. ModelMuse will determine what the stress periods ought to be based on the times entered by the user. Thus, the specification of the boundary conditions is independent of the time discretization specified by the user. The Time-Variant Specified Head (CHD) package in MODFLOW is different from other packages in that once a cell has been specified as a CHD cell, it can not be converted back to a cell that does not have a CHD boundary in it. (This is determined by how the CHD package is implemented in MODFLOW.)

Model Features in PHAST

In PHAST models, Model Features are used to specify the specified-head, flux, leaky, river, and well boundary conditions. The user must specify starting times for each boundary condition. The values specified for each time apply until the model run is terminated or the values are overridden by values for a later time.

Model Features in MODFLOW

In MODFLOW models, Model Features are used to define some or all of the spatial data in the following packages as well as the IFACE variable and the starting particle distribution in MODPATH.

- CHD: Time-Variant Specified-Head package
- CHOB: Specified-Head Flow Observation package
- DRN: Drain package
- DROB: Drain Observation package
- DRT: Drain Return package
- ETS: Evapotranspiration Segments package
- EVT: Evapotranspiration package
- GHB: General-Head Boundary package
- GBOB: General-Head-Boundary Observation package
- HFB: Horizontal Flow Barrier package
- HOB: Head Observation package
- LAK: Lake package
- RCH: Recharge package
- RES: Reservoir package
- RIV: River package
- RVOB: River Observation package

- SFR: Stream-Flow Routing package
- UZF: Unsaturated-Zone Flow package
- WEL: Well package

The user specifies a start and end time for the boundary condition. The user does not need to define boundary conditions by stress period because ModelMuse can determine the appropriate stress periods based on the times specified by the user.

Comparison of Objects and Shapefiles

Users of Geographic Information Systems (GIS) may find some similarities between Objects in ModelMuse and coverages in a GIS. There are, however, significant differences as well. The precise characteristics of a GIS coverage may vary among implementations in different software and formats. Shapefiles (Environmental Systems Research Institute, Inc., 1998), one example of a GIS coverage, resemble Objects in that they comprise points, polylines, polygons (and some other shapes) associated with attributes. Unlike objects, however, all the shapes in a Shapefile are associated with the same kinds of attributes. In contrast, the data sets associated with one object are not necessarily the same as the data sets associated with another object. The values of the attributes associated with a shape in a Shapefile are simple types and thus do not vary from place to place within a shape. With objects, the formula for a data set is evaluated at different locations and can give a different value at each such location. Both objects and shapes have strict rules regarding what is and is not a valid geometry; however, the rules are not the same for both of them. In shapes with multiple sections, for example, no section can overlap any other section, whereas in Objects, overlapping sections are allowed. Another difference between shapes and Objects is that all the shapes in a Shapefile must be of the same type; combinations of points, polylines, and polygons in the same Shapefile are not allowed. Objects have no such restriction. In summary, shapes in a Shapefile tend to be much more homogeneous than are Objects in ModelMuse.

Initial Dialog Boxes

When the user first starts ModelMuse, the **Start-Up** dialog box is displayed. If the user chooses to create a new model in the **Start-Up** dialog box, the **Initial Grid** dialog box is displayed. After both these dialog boxes are closed, the main window of ModelMuse is displayed (fig. 4).

The **Initial Grid** dialog box is used to specify the grid for a new ModelMuse project either when first starting ModelMuse or by selecting **File|New**. Its appearance varies slightly depending on whether the user is creating a new MODFLOW model or a new PHAST model.

Clicking the **Finish** button will create a grid with the dimensions and location specified. If the **No grid** button is clicked instead, a new project will be created without a grid. The grid can be created later using the methods described in Generating Grids.

Initial Grid Dialog Box for PHAST

With PHAST, the user specifies the dimensions of the grid (the number of columns, rows, and layers of nodes) and the spacing between nodes in the column, row, and layer directions. The user must also specify the X, Y, and Z coordinates of the grid origin. The grid origin is the location of the node in the first column, row, and layer. Columns are numbered from left to right. Rows are numbered from front to back. Layers are numbered from bottom to top. Thus the grid origin is the node at the left, front, bottom corner of the grid. The grid angle must also be specified. The angle is measured in degrees counterclockwise from the X-axis. The vertical exaggeration may also be specified. If it is not specified, a reasonable default value will be calculated.

Initial Grid Dialog Box for MODFLOW

With PHAST, the user specifies the dimensions of the grid (the number of columns, rows, and layers) and the width of the columns and rows. The user must also specify the X, Y, and Z coordinates of the grid origin. The grid origin is the location of the node in the first column, row, and layer. Columns are numbered from left to right. Rows are numbered from back to front. Layers are numbered from top to bottom. Thus the grid origin is the node at the left, back, top corner of the grid. The grid angle must also be specified. The angle is measured in degrees counterclockwise from the X-axis. The vertical exaggeration may also be specified. If it is not specified, a reasonable default value will be calculated. The user also specifies the elevation of the top of the model, the names of the aquifers, and the elevations of the bottoms of those aquifers.

Main Window

The main window of ModelMuse has several parts, as listed below and shown in figure 5:

- Main Menu and Buttons,
- Top, Front, and Side Views of the model,
- 3–D View of the model, and
- Status Bar.

The menu and buttons are described in Main Menu and Buttons. The other parts are described in the following sections.

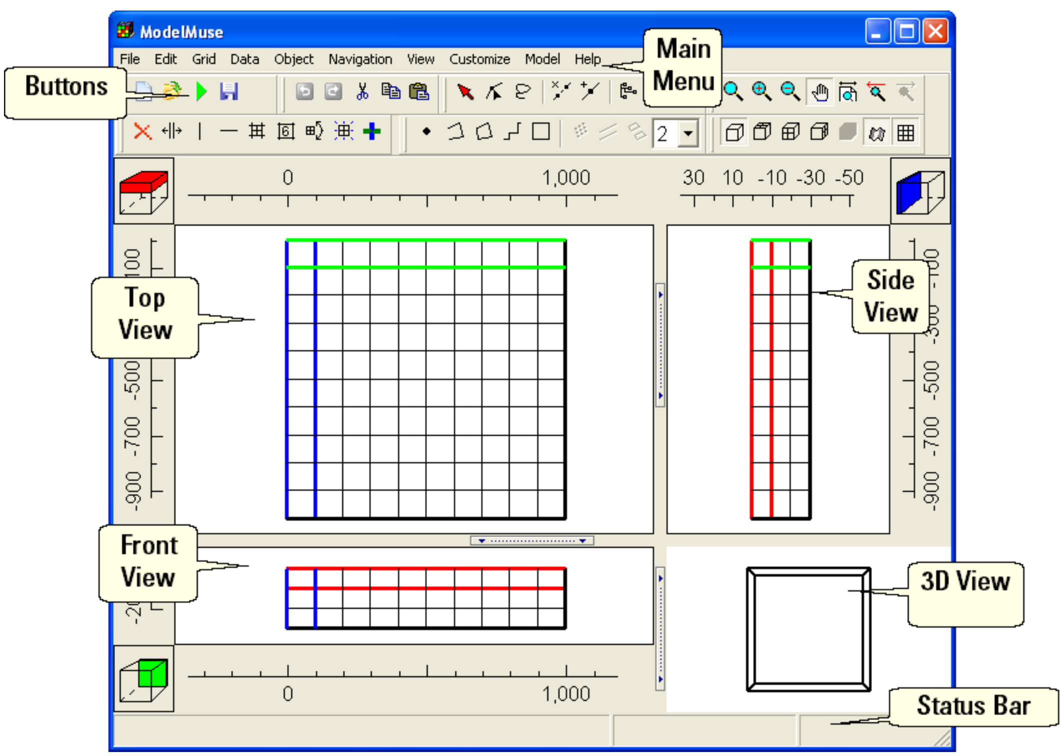


Figure 5. The main window of ModelMuse.

Top, Front, and Side Views

The main ModelMuse window includes four panes. Three of these panes contain the top, front, and side views of the model. The other pane is a 3–D view of the model. Each pane can be resized by clicking on the space between the panes, moving the mouse while holding the mouse button down, and releasing the mouse button at the new position.

The top, front, and side views of the model are each composed of several parts (fig. 6):

- Selection Cube,
- Rulers, and
- Working Area.

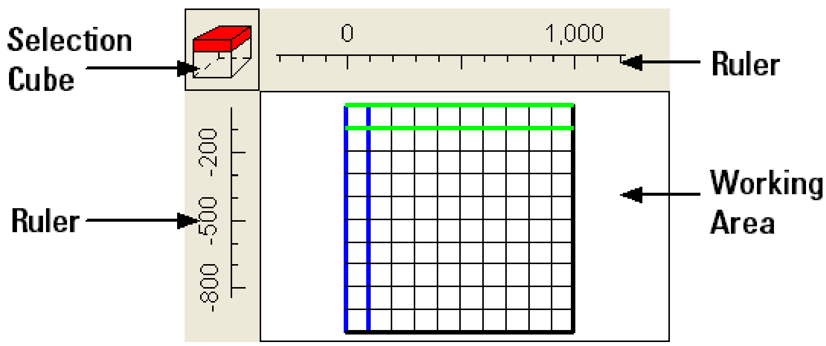


Figure 6. The parts of the top, front, or side views of the model.

The Selection Cube

The **Selection Cube** (fig. 7) shows the selected column, row, or layer. It can also be used to change the selected column, row, or layer.

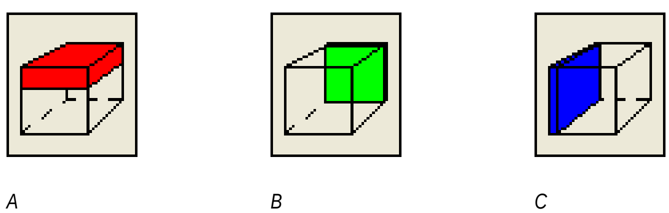


Figure 7. The top, front, and side Selection Cubes.

- To change the selected column, row, or layer with the Selection Cube, click on the Selection Cube. The selected column, row, or layer will move by one node or element toward the position that was clicked.
- If the Shift key on the keyboard is held down when the mouse button is clicked, the selected column, row, or layer will move by 10 nodes or elements toward the position that was clicked, or to the end of the grid, whichever is less.
- If the Ctrl key on the keyboard is held down when the mouse button is clicked, the selected column, row, or layer will move to the position that was clicked.
- If the left mouse button is held down while the cursor is on the Selection Cube, the selected column row or layer will start to move toward the cursor position after a wait of one second. It will then move rapidly toward the cursor.

The Ruler

The **Ruler** (fig. 8) is used to show the position of the model. The format of the numbers on the Ruler can be changed by double clicking on a Ruler or selecting **Customize|Ruler Format...**

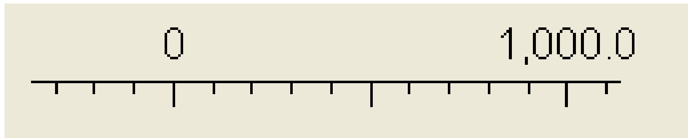


Figure 8. Ruler in ModelMuse.

The Working Area

The **Working Area** is used to display and edit the model. All objects are created and edited in the **Working Area**. It is also possible to edit the grid in the Working Area. The **Zoom**, **Zoom In**, **Zoom Out**, and **Pan** buttons allow the user to navigate in the Working Area. The menu command **Navigation|Restore Default 2D View** will zoom in or out so that most or all of the grid is visible in the top, front, and side views. See the Grid, Object, and View menu items under Main Menu and Buttons for more details.

Three-Dimensional View

The 3–D view of the model (fig. 9) allows the user to view (but not edit) the model. A number of items in the **View** menu change the appearance of the 3–D view. The following mouse actions can be used to navigate in the 3–D view.

- To rotate the 3–D view, click in the 3–D view and drag with the mouse.
- To pan the 3–D view, hold down the shift key while dragging with the mouse.
- To change the magnification of the 3–D view, hold down the right mouse key and click in the 3–D view. Then, while holding the right mouse key down, move the mouse up or down.

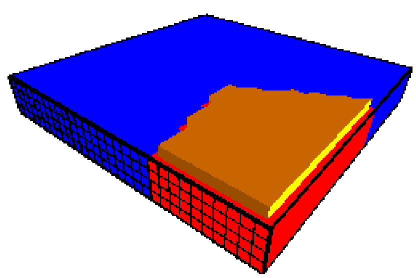


Figure 9. A 3–D view of a model in ModelMuse.

Hints and the Status Bar

When the mouse is held briefly over a menu item or button in the main ModelMuse window, a hint will appear on the status bar that briefly describes the function of the menu item or button. In addition, with buttons, a shorter version of the “hint” will appear in a small window in front of the button. The hint will remain visible for a short time and then disappear.

Hints that are related to the **Working Area** are also displayed on the status bar. When the cursor is moved over the **Working Area**, the coordinates of the cursor are displayed on the left panel of the status bar. When the cursor is over the grid, the column, row, and/or layer number is displayed on the middle panel of the status bar. When the grid is colored, the value of the data set used to color the grid in the current node or element is displayed on the right-hand panel of the status bar along with a description of how the value was assigned. The value and description may also be viewed in the **Grid Value** dialog box (**Data|Show Grid Values**).

To resize the individual panels on the status bar, click on the dividers between the panels and drag with the mouse.

See also “Hint Display Time Dialog Box” in the help system (**Customize|Hint Display Time**).

Creating, Selecting, and Editing Objects in ModelMuse

The five types of objects in ModelMuse are:

- Points ,
- Polylines ,
- Polygons ,
- Straight-Lines , and
- Rectangles .

Each object consists of one or more vertices and has properties associated with it. The properties of an object determine how it is used to specify spatial properties of the model. Each object is associated with one of the three views of the model (top, front, or side).

Creating Objects

Objects are created by drawing them on the top, front, or side Working Areas. The following sections describe how to create each type of object.

Points

Point objects have only a single vertex. To create a point object, the user does the following:

1. The user either selects **Object|Create|Point** or clicks on the **Create point object** button .
2. The user moves the cursor to the location on one of the views of the model where a point object is desired.
3. The user clicks the mouse button. A point object will be created at the cursor location. The **Object Properties** dialog box will appear. The **Object Properties** dialog box is used to specify the properties of the object.

Typical uses for point objects are to define the elevations of the tops or bottoms of geologic units, and to assign point values for hydraulic properties that will be defined using interpolation, well boundary conditions, and head observations.

Polylines

Polyline objects have two or more vertices. To create a polyline object, the user does the following:

1. The user either selects **Object|Create|Polyline** or clicks on the **Create polyline object** button .
2. The user moves the cursor to the location on one of the views of the model where the first vertex of the polyline object is desired.
3. The user clicks the left mouse button. The first vertex of the polyline object will be created at the cursor location.
4. To create additional vertices, the user continues clicking on the same view of the model.
5. When all the desired vertices have been created, the user double-clicks at the last vertex or presses the “Enter” key on the keyboard to complete the polyline.

Pressing the escape or delete key while creating an object deletes the last created vertex of that object. As with points, the **Object Properties** dialog box is used to specify the properties of the object.

Typical uses for polylines are to define linear features, such as rivers and boundary conditions at the edges of the model.

Polygons

Polygon objects have four or more vertices. The last vertex is always at the same location as the first vertex, so a polygon with four vertices is a triangle. To create a polygon object, the user does the following:

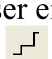
1. The user either selects **Object|Create|Polygon** or clicks on the **Create polygon object** button .
2. The user continues adding vertices as described with Polylines.
3. When the polygon is complete, a final vertex will be added at the location of the first vertex to close the polygon.

Pressing the escape or delete key while creating an object deletes the last created vertex of that object. As with points, the **Object Properties** dialog box is used to specify the properties of the object.

Typical uses for polygons are to define zones with differing media properties or boundary conditions such as lakes.

Straight Lines

Straight-line objects are a special case of polylines in which all the line segments in the polyline are parallel to one of the edges of the grid. To create a straight-line object, the user does the following:

1. The user either selects **Object|Create|Straight Line** or clicks on the **Create straight-line object** button .
2. The user continues adding vertices as described with Polylines.

Pressing the escape or delete key while creating an object deletes the last created vertex of that object. As with points, the **Object Properties** dialog box is used to specify the properties of the object.

Typical uses of straight-line objects are the same as for polylines; that is, to define linear features, such as rivers and boundary conditions at the edges of the model. Straight-line objects are more frequently used in purely hypothetical models where the model is kept very simple rather than in models of actual systems.

Rectangles

Rectangles are a special case of polygons that have four edges that are parallel to the grid. To create a rectangle, the user does the following:

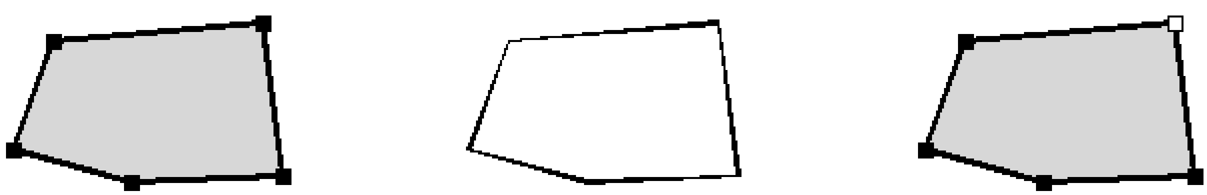
1. The user either selects **Object|Create|Rectangle** or clicks on the **Create rectangle object** button .
2. The user moves the cursor to the location on one of the views of the model where the first vertex of the rectangle object is desired.
3. The user clicks the mouse button again at the opposite corner of the rectangle.

As with points, the **Object Properties** dialog box is used to specify the properties of the object.

Typical uses of rectangle objects are the same as for polygons; that is, to define zones with differing media properties. Rectangle objects are more frequently used in purely hypothetical models rather than in models of actual systems.

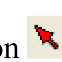
Selecting Objects

To edit, move, or delete an object, the object must first be selected. Selected objects can be distinguished from objects that are not selected because the lines used to draw the selected objects are thicker, the vertices of the selected objects are shown as squares, and, if they are polygons, they are shaded. However, point objects are also shown as squares even if they are not selected. Selected point objects are shown as solid squares whereas point objects that are not selected are shown as hollow squares. In other types of objects, the vertices are shown as solid squares when the whole object is selected. If an individual vertex is selected, it is shown as a hollow square. The object in figure 10*A* is selected; the object in figure 10*B* is not. The object in figure 10*C* has one of its vertices selected.



A *B* *C*

Figure 10. Appearance of (*A*) selected object, (*B*) nonselected object, and (*C*) an object with a selected vertex.

Objects can be selected in several ways. One way is to click on the **Select objects** button  and then click on the object. Another way is to click and hold the left mouse button down somewhere on one view of the model that is not on any object and then drag the mouse. A rectangle will be drawn starting where the mouse was clicked down. When the mouse button is released, any objects that are entirely inside the rectangle will be selected. Another option is to click on the **Lasso** button . Then the user clicks down on one view of the model and moves the mouse. A line will follow the mouse. When the user releases the mouse, any objects that are entirely inside the line will be selected.

Usually, selecting one object causes any object that was previously selected to become nonselected; however, if the Shift key is held down while selecting an object, then the objects that would normally be selected are toggled between selected and nonselected and other objects remain selected.

The Ctrl key also modifies how selection occurs when the **Select objects** button is down but not when the **Lasso** button is down. The user can use the Ctrl key to select objects beneath another object. If several objects are on top of one another, the user can click on the objects with the Ctrl key down and if none of the objects in the group is selected, the uppermost one will be selected. If the user clicks again with the Ctrl key down, the top object will be deselected and the next one down will be selected. If the user clicks again with the Ctrl key down, the object below that will be selected. The Shift and Ctrl key can be combined.

Individual vertices of an object can be selected as well as whole objects. To select individual vertices, the user first selects an object and then clicks on the **Select vertices** button. Next, the user clicks on the desired vertex to select it. To select additional vertices, the user clicks on them while holding down the Shift key. Vertices can only be selected on objects that are selected. The user can also click down in an area away from any vertices of a selected object, drag the mouse, and then release it to select vertices that are inside the rectangle defined by the locations where the user clicked down with the mouse button and where the user released it. All the vertices outside the rectangle will become nonselected (unless the Shift key is down). If the user holds down the Shift key while clicking on a vertex, the vertex will be toggled from selected to nonselected or the reverse.

A number of dialog boxes also allow the user to select objects. See “Search for Objects Dialog Box”, “Show or Hide Objects Dialog Box”, “Select Objects by Name Dialog Box”, and “Go To Dialog Box” in the help system.

Editing Objects

Objects can be edited in a number of ways. Objects can be deleted or moved. Individual vertices in objects can be inserted, moved, or deleted. Edges of objects can be deleted. New sections can be added to an object. The order of the objects can be changed. Finally, the properties of objects can be changed.

To delete an object, select it and press the delete key on the keyboard. To delete individual vertices, select the vertices and press the delete key. To delete an edge of an object, click on the **Delete segment** button. Then click on an edge of an object to delete that edge. If deleting an edge will split the object into two separate pieces, each piece will become a separate section of the same object.

One way to move objects or vertices is to select them, hold down the mouse button with the mouse curser on or inside the object, and move the mouse before releasing the mouse button. Another way to move an object or an individual vertex is to double click on the object. Then, on the tab labeled **Vertices** in the **Object Properties** dialog box, type in new values for the coordinates of the vertices.

To insert a vertex, first click on the **Insert vertex** button. Then click on the edge of an object to insert a vertex at the position where the mouse was clicked. If any object is selected, this procedure will only insert a vertex in the selected object. If no object is selected, this procedure will insert a vertex into the object on which the user clicks.

To change the order of objects, select one or more objects and right-click on them. Select **To Front**, **To Back**, **Forward One** or **Back One** in the pop-up menu to change the order of the objects. It is also possible to select **Object|Rearrange Objects…** The **Rearrange Objects** dialog box will appear. In it, the user can drag objects to new positions. When assigning values to data sets, the order of the objects is important because each object that assigns values at enclosed or intersected locations replaces values assigned previously by the default formula or by a previous object. The order of the objects is important for interpolation only if more than one object is at the same location.

To edit the properties of one or more objects, select them and then double-click on one of them. The **Object Properties** dialog box will appear, and the properties can be edited. The user can also display the **Object Properties** dialog box to edit the properties of a single object by displaying the **Show or Hide Objects** dialog box and double-clicking on the name of the object.

To add additional sections to an existing object, the user first selects the object and then clicks the **Add point sections** , **Add polyline sections** , or **Add polygon sections** button. Next the user clicks on the view of the model containing the object at the location where a new vertex is desired. When adding polylines or polygons to an existing object, double-clicking or pressing the “Enter” key on the keyboard finishes the polyline or polygon section.

Generating Grids

A grid can be generated in four ways.

1. When a new model is first created, a uniform grid can be specified.
2. The commands **Grid|Specify Grid Angle…** and **Grid|Specify Grid Lines…** can be used to specify the grid numerically.
3. A grid can be drawn and rotated with the mouse.
4. The user can use objects to define where the grid should be and then use the **Grid|Generate Grid...** command to create it.

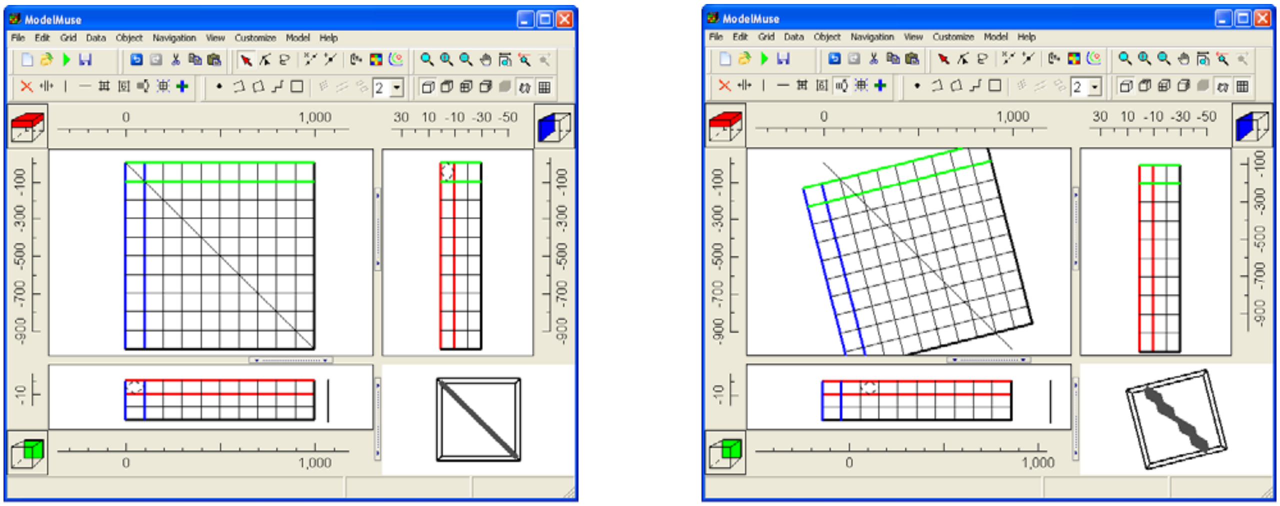
These four options are explained in greater detail in the following sections.

Specifying a Uniform Initial Grid

When the user chooses to create a new model, the Initial Grid dialog box gives the user the opportunity to specify a grid with uniformly spaced columns and rows. With PHAST, the layers will also be uniformly spaced. With MODFLOW, the layers will be horizontal but the layer thicknesses will be specified by the user.

Specifying a Grid with Numbers

If precise control over the position of the grid is desired, the menu items **Grid|Specify Grid Angle…** and **Grid|Specify Grid Lines…** can be used to locate the grid at an exact position. Normally, it is best to specify the grid angle first. When the grid angle is changed, the grid is rotated relative to its own center. The positions of the grid lines are measured relative to the origin of the coordinate system so that when the grid is rotated, those positions change, as illustrated in the front and side views of the model in figure 11. The objects are not rotated when the grid is rotated (fig. 10). See “Grid Angle Dialog Box” and “Grid Spacing Dialog Box” in the help system for more information.



A *B*

Figure 11. Unrotated (*A*) and rotated (*B*) grid with unmoved objects on the top and front views.

Drawing the Grid

If the user selects **Grid|Add Vertical Grid Line** or **Grid|Add Horizontal Grid Line** and then clicks on the top, front, or side view of the model, a horizontal or vertical grid line will be drawn at the mouse position. Layer boundaries can be drawn this way in PHAST models, but with MODFLOW models, the **MODFLOW Layer Groups** dialog box must be used to define the layers. The grid can be rotated around its center by selecting **Grid|Drag to Rotate** and then dragging with the mouse on the top view of the model. Other menu items that are useful in drawing the grid include **Grid|Delete Grid Line**, **Grid|Move Grid Line**, **Grid|Subdivide Grid Elements**, **Grid|Set Width**, and **Grid|Smooth Grid...** See “Editing the Grid” in the help system for more information on these menu items.

Using Objects to Specify the Grid

Objects (except for point objects) can be used to set the size of elements (PHAST) or cells (MODFLOW) in a grid. Multiple objects can be used in determining the grid location and node spacings. An object drawn on the top view of the model can be used to specify the column and row widths. Polygons drawn on the top view of the model that are used to specify the grid element or cell size also specify the grid location. The grid will be drawn to completely enclose the polygon or polygons. With PHAST, an object drawn on the front or side view of the model specifies the layer height and the vertical extent of the grid. For an object to be used to specify the grid, the **Use to set grid element\cell size** checkbox in the **Object Properties** dialog box must be checked. When it is checked, the user can enter the desired size in the **Grid element\cell size** edit box. The **Object Properties** dialog box appears when an object is created or when the user double-clicks on an object.

In both PHAST and MODFLOW, having an area with a refined grid can be useful where results need to be more accurate. To support this, ModelMuse allows the user to specify overlapping objects that specify different element sizes. The final sizes of the elements will be determined by the smallest element size specified for a region. In MODFLOW (but not PHAST) it is important that the contrast in element size in adjacent elements not be too large. When the user creates a grid using objects, the **Grid smoothing criterion** in the Generate Grid dialog box allows the user to specify the maximum ratio of lengths between nodes for adjacent rows, columns, or layers that will be accepted. The default value is 1.2, which is well below the usual limit of 1.5.

The following steps illustrate the creation of grids by using objects.

1. The user draws one or more polygons on the top view of the model to define the grid location and the size of the elements (PHAST) or cells (MODFLOW). For PHAST, the user must also draw a line on the front or side view of the model (fig. 12). In MODFLOW, the vertical discretization is defined in the **MODFLOW Layer Groups** dialog box. (To create objects, see “Creating, Selecting, and Editing Objects in ModelMuse” in this report and the **Object** menu item.) In this example, the size of the elements are set in the **Object Properties** dialog box to 1,000 m in the horizontal direction (top view, fig. 12) and 10 m in the vertical direction (front view, fig. 12).

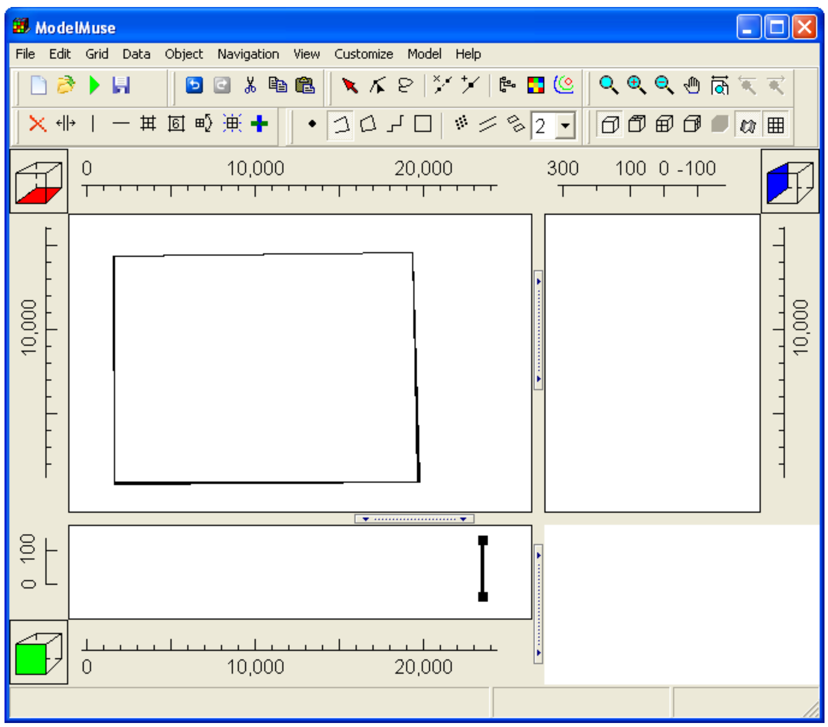


Figure 12. Two objects used to define the position of the grid.

2. The user selects **Grid|Generate Grid...**  and the **Generate Grid** dialog box appears (fig. 13). When the user clicks on the **OK** button, the grid is created (fig. 14).

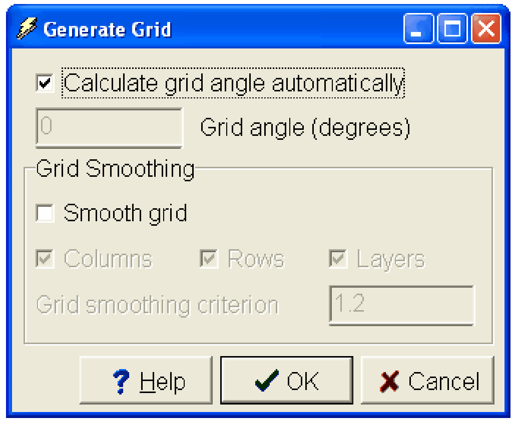


Figure 13. The Generate Grid dialog box.

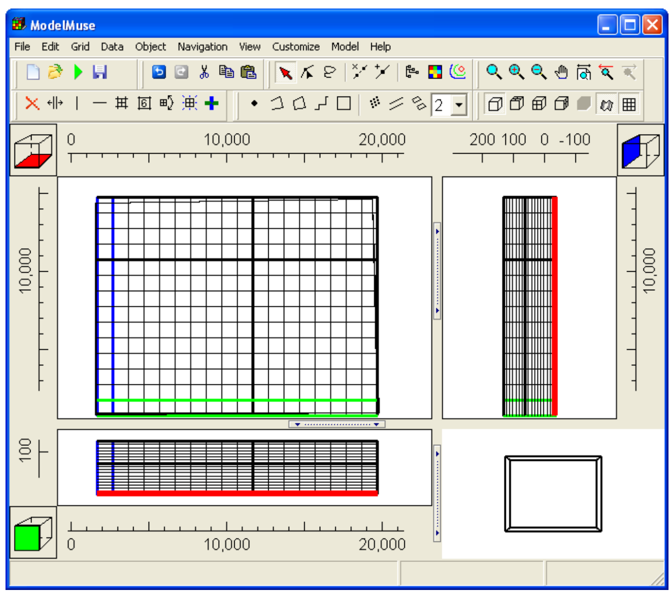


Figure 14. Grid and objects.

3. It may be desirable to have a finer grid in some regions than others. To do this, additional objects can be used to specify a zone where a fine grid will be used (fig. 15). In this example, the object specifies an element size of 200 m in the **Grid element size** edit box of the **Object Properties** dialog box.
4. After choosing **Grid|Generate Grid...** again, a new grid is created with a smaller element size in the area enclosed by the object (fig. 15).

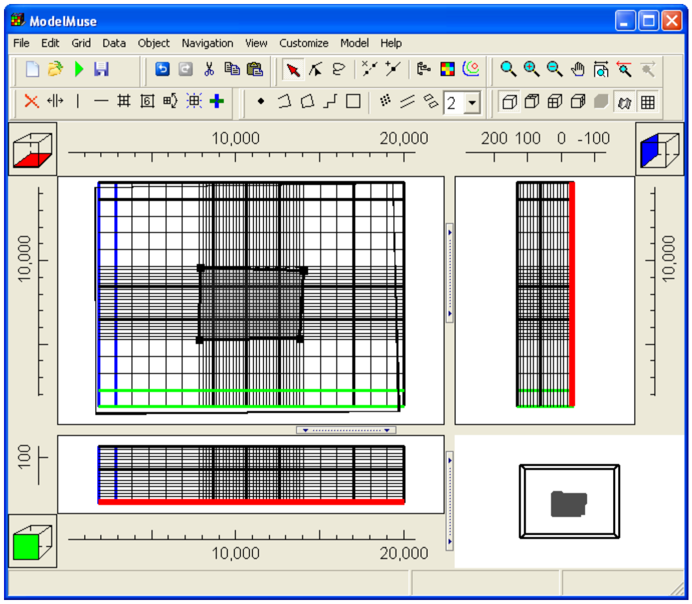


Figure 15. Grid with region with smaller elements specified by polygon object.

5. In some cases, it may be desirable to have a gradual transition in element size. A size transition can be achieved by checking the **Smooth grid** checkbox in the **Generate Grid** dialog box (fig. 16). When it is checked, the widths of the columns, row, or layers will be adjusted so that the maximum ratio of the widths of adjacent cells is less than or equal to the grid smoothing criterion. An example of such a grid is shown in figure 17. See "Editing the Grid" in the help system for more information on grid smoothing.

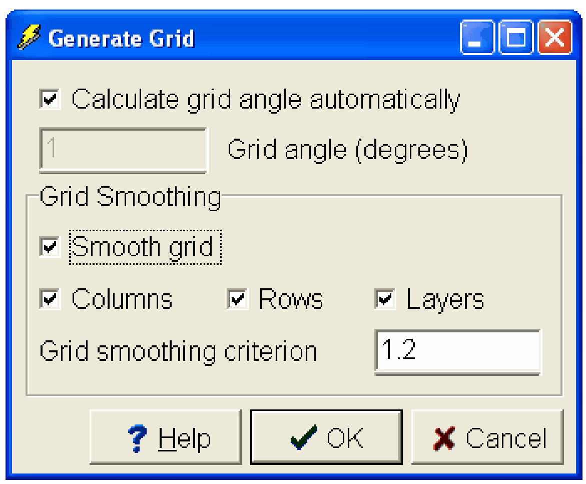


Figure 16. Generate Grid dialog box with grid smoothing activated.

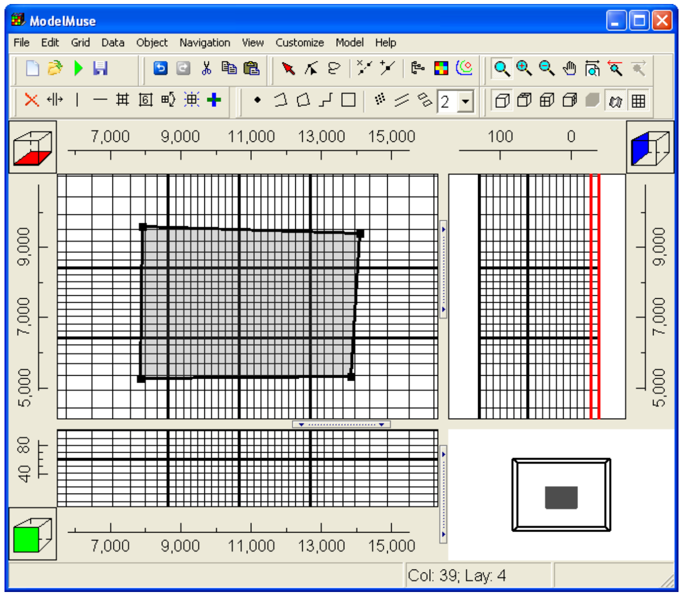


Figure 17. Grid generated with grid smoothing.

Interpolation Methods

The interpolation method of a data set is used to determine how values should be interpolated among a group of objects. Interpolation can only be used for 2–D data sets. Only one of the PHAST data sets, Initial_Water_Table, is a 2–D data set. All the data sets defining the boundaries between layers in MODFLOW are 2–D data sets. In addition, the user can define his or her own 2–D data sets and use interpolation in them. One appropriate use of such 2–D data sets in PHAST would be to define the tops and bottoms of geologic units.

Six interpolation algorithms are available in ModelMuse: **Nearest**, **Nearest Point**, **Inv. Dist. Sq.** (Inverse Distance Squared), **Triangle Interp.** (Triangle Interpolation), **Fitted Surface**, and **Point Inv. Dist. Sq.** (Point Inverse Distance Squared).

The **Nearest** interpolation method (fig. 18*A*) works by determining the object that is closest to the location where the data set in question is being evaluated. Then the formula of that object is evaluated at that location. If all the objects used with the method are point objects, the **Nearest Point** interpolation method produces the same result more quickly (fig. 19) so the **Nearest** interpolation method is generally used for line or polygon objects.

The **Nearest Point** interpolation method (fig. 18*B*) is similar to the **Nearest** interpolation method except that only the vertices of objects are considered, rather than the lines connecting the vertices. If only point objects are used, the result is identical. The **Nearest Point** interpolation method uses an algorithm that is faster (fig. 19) than the **Nearest** interpolation method when the number of points is greater than several hundred. The **Nearest Point** interpolation method is the fastest of all the interpolation methods (fig. 19).

With the **Inv. Dist. Sq.** interpolation method (fig. 18*C*), the formula for each object is evaluated at the location under consideration. The final value is a weighted average of these values. The weights are the inverse of the distance squared from the location to the closest point on each respective object. The **Inv. Dist. Sq.** interpolation method may only be used with data sets containing real numbers. Because the **Inv. Dist. Sq.** interpolation method is slow (fig. 19), it generally is not used with more than a few hundred objects. If the formulas for all the objects are numbers and all the objects are point objects, the **Point Inv. Dist. Sq.** interpolation method gives the same results more quickly (fig. 19).

With the **Point Inv. Dist. Sq.** interpolation method (fig. 18*D*), the formula for each vertex for each object is evaluated at its own location. The final value is a weighted average of these values. The weights are the inverse of the distance squared from the location of interest to the vertex. The **Point Inv. Dist. Sq.** interpolation method may only be used with data sets containing real numbers. If the formulas for all the objects are numbers and all the objects are point objects, the **Point Inv. Dist. Sq.** interpolation method gives the same results as the **Inv. Dist. Sq.** interpolation method but does so more quickly (fig. 19).

The **Fitted Surface** interpolation method (fig. 18*E*) evaluates the formula for each object at the location of each vertex on the object. It then creates an unconstrained Delaunay triangulation of all the vertices (the same one as in **Triangle Interp.)**. A piece-wise continuous function of the locations is fitted through the data values and the function is used to assign values at each location of interest. This method was created by Renka (1996a, b). **The Fitted Surface** interpolation method may only be used with data sets containing real numbers. Interpolated values returned by **Fitted Surface** may be higher than the highest value at any data point or lower than the lowest value at any data point. The **Fitted Surface** interpolation method is a relatively fast interpolation method (fig. 19).

The **Triangle Interp.** interpolation method (fig. 18*F*) evaluates the formula for each object at the location of each vertex on the object. It then creates an unconstrained Delaunay triangulation of all the vertices. If a location where a value is needed is inside one of the triangles, the value assigned to that location will be calculated by fitting a plane to the three points of the triangle and determining the height of the plane at the location of interest. This method was created by Renka (1996a, b). The **Triangle Interp.** interpolation method may only be used with data sets containing real numbers. Interpolated values returned by **Triangle Interp.** are never higher than the highest value at any data point or lower than the lowest value at any data point. The **Triangle Interp.** interpolation method is a relatively fast interpolation method (fig. 19).

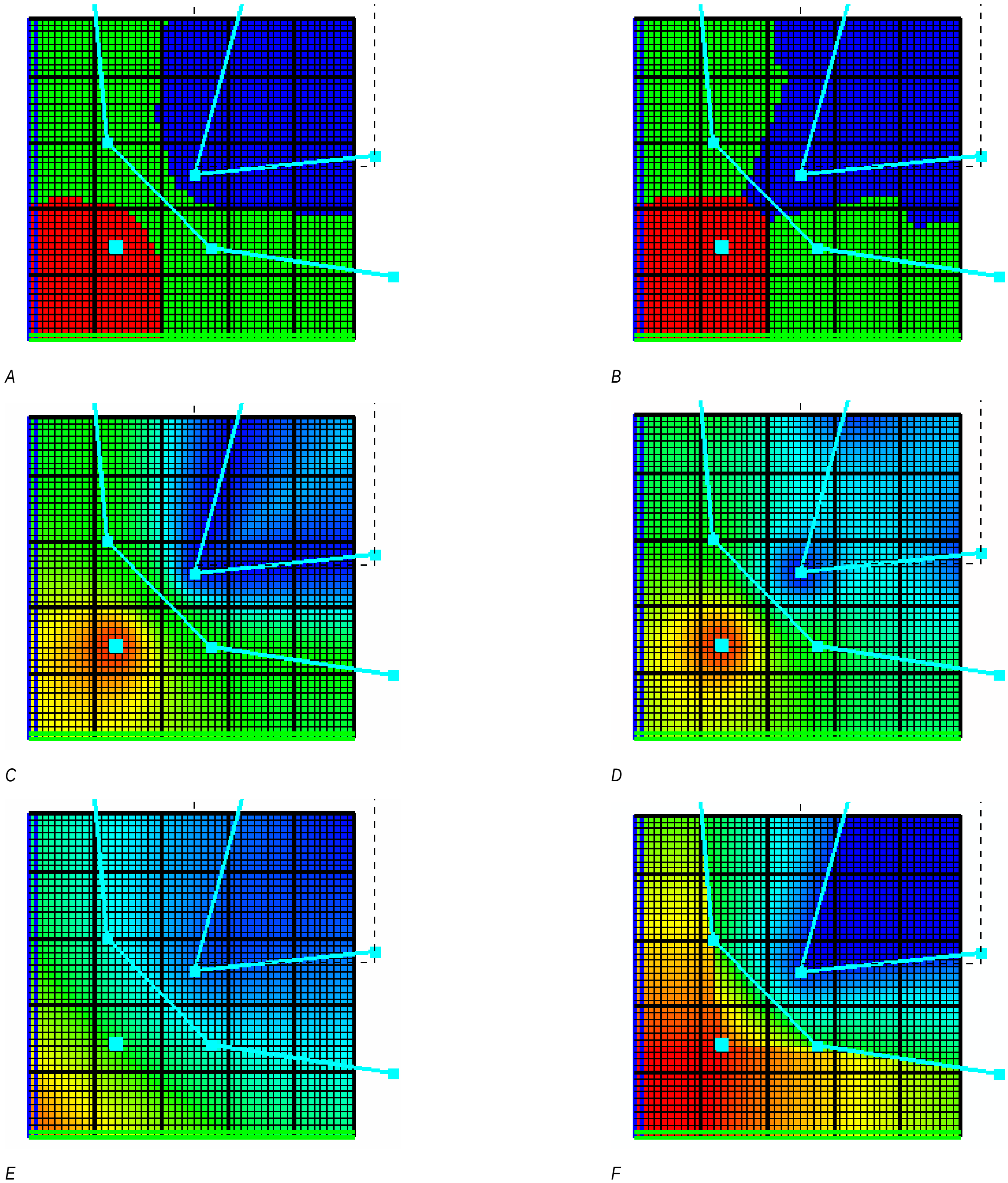


Figure 18. Interpolation methods. The points and lines are objects with different values. The figures show how the interpolated values vary among interpolation methods when applied to the same data—(*A*) **Nearest** interpolation method, (*B*) **Nearest Point** interpolation method, (*C*) **Inv. Dist. Sq.** interpolation method, (*D*) **Point Inv. Dist. Sq.** interpolation method, (*E*) **Fitted Surface** interpolation method, and (*F*) **Triangle Interp.** interpolation method.

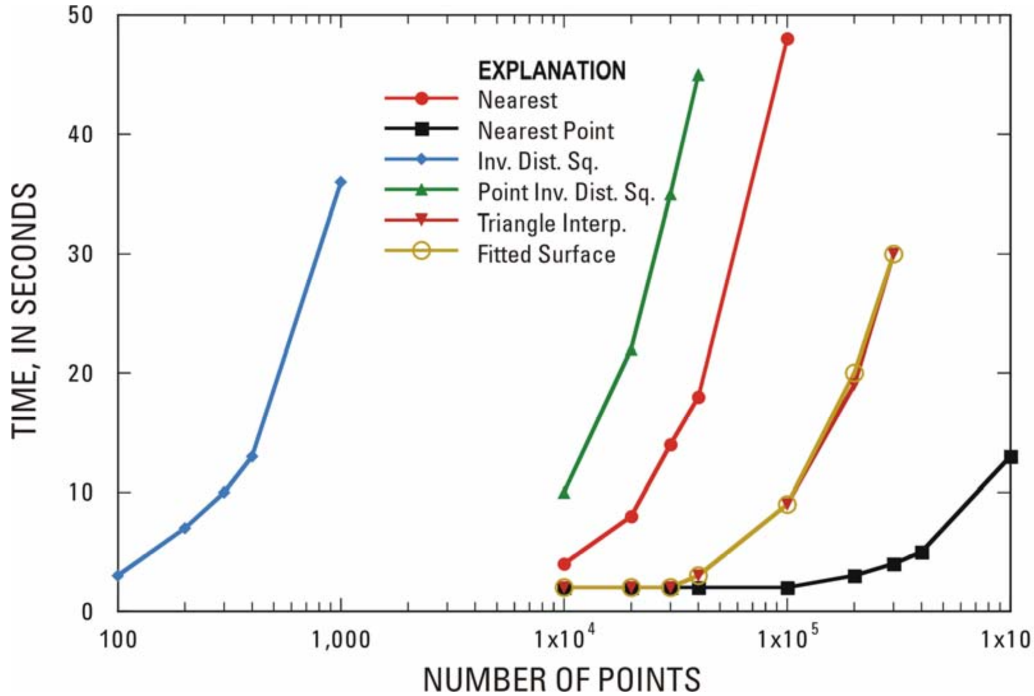


Figure 19. Time required to assign values to a grid with 10,000 cells as a function of the number of points using the various interpolation methods. Experiments were performed using a 2.0 GHz CPU.

PHAST–Style Interpolation

PHAST has a built-in interpolation method using the grid coordinate system that can be applied to some properties using keyword data blocks of the following types (Parkhurst and others, 2004):

- CHEMISTRY_IC,
- FLUX_BC,
- HEAD_IC,
- LEAKY_BC,
- MEDIA, and
- SPECIFIED_HEAD_BC.

In a PHAST-style interpolation, the user specifies a coordinate direction (X, Y, or Z), two distances, and two values. The coordinate direction is relative to the grid so if the grid is at an angle these are actually the X', Y' and Z directions (fig. 20). If the X', Y', or Z coordinate of the current node is less than or equal to the first distance, the first value (value 1) is used. If the X', Y', or Z coordinate of the current node is greater than or equal to the second distance, the second value (value 2) is used. For intermediate distances, linear interpolation between the two values is used.

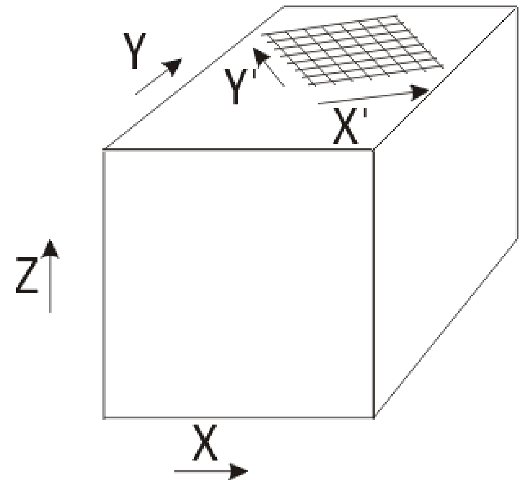


Figure 20. Global and grid coordinate systems in ModelMuse.

A related capability in PHAST is the ability to specify mixtures between two chemical compositions. In PHAST-style mixtures, the user specifies two values that represent chemical compositions and a series of proportions (between 0 and 1). The proportion represents the fraction of **Value 1** in the mixture.

The following controls are used to specify the values needed for PHAST-style interpolation.

- **Interp. dir. or mixture**: Interp. dir. or mixture indicates the direction in which the interpolation will take place. For some data sets, the user can specify **Mix**. If **Mix** is specified, the mixture formula is used to specify the fraction of **Value 1** in the mixture.
- **Distance 1** and **Distance 2**: **Distance 1** is the lower end of the distances used for interpolation. **Distance 2** is the upper end of the distances used for interpolation. If the position of a cell or element is between **Distance 1** and **Distance 2**, the value applied to a cell will be interpolated between **Value 1** and **Value 2**. Otherwise, either **Value 1** or **Value 2** will be applied depending on which end of the range of distances is closest to the position of the cell or element.
- **Value 1** and **Value 2**: **Value 1** and **Value 2** are the two values that will be used for interpolation.
- **Mixture formula: Mixture formula** is the fraction of **Value 1** to use in the mixture of **Value 1** and **Value 2**. The **Mixture formula** should range between 0 and 1. If the **Mixture formula** is outside the range of 0 to 1, a value of either 0 or 1 will be used depending on which is closest to the value of the mixture formula.

PHAST-style interpolation in ModelMuse can only be applied to data sets in PHAST that allow PHAST-style interpolation. This includes all the data sets directly used in the PHAST transport data file except Active, Print_Chemistry, and Print_XYZ_Chemistry. PHAST-style interpolation cannot be used with data sets created by the user. However, the user could specify a formula that would have the same effect as PHAST-style interpolation. Either of the following will emulate PHAST-style interpolation when applied to a real-number data set:

- If((X_Prime < 1000.), 25., If((X_Prime > 10000.), 60., ((((X_Prime - 1000.) / 9000.) * 35.) + 25.)))
- MultiInterpolate(X_Prime, 25, 1000, 60, 10000)

In these formulas, the interpolation direction is X', the two distances are 1,000 and 10,000, and the two values are 25 and 60.

Data sets that store integers and whose values are set by PHAST-style interpolation or mixtures should not be used in formulas for other data sets because the value used in the formula will be rounded to the nearest integer rather than representing a mixture between the two end member compositions. The formulas used in ModelMuse are strictly mathematical and are unable to interpret those numbers as chemical compositions.

Formulas

Formulas are used to assign values to nodes or elements and to define the geometry of 3–D objects. The simplest formula is simply a numerical value such as "100." or "1." In a new model, if the user selects **Data|Edit Data Sets...**, the **Data Sets** dialog box will appear and it can be seen that the default formulas for **Kx** is "100." This means that, in the absence of any other method (see Assigning Values to Data Sets) for assigning a value to an element, **Kx** for all elements will be 100. By using a slightly more complicated formula, it is possible to see the power of formulas. Here is an example:

- In the tree control on the left select "Required|Hydrology|Kz"
- In the **Default Formula** box, type "Kx/100" (without the quotation marks).

In the absence of any other method for assigning a value of **Kz** to an element, **Kz** for all elements will be equal to the value of **Kx** for that element divided by 100. Now by specifying the spatial distribution of **Kx**, the spatial distribution of **Kz** is also specified.

If the user clicks the button labeled **Edit Formula**, the **Formula Editor** will appear. The **Formula Editor** can help set up complex formulas correctly. See the "**Formula Editor**" in the help system for more information.

When a Formula is applied to a data set or boundary condition in the **Data Sets** tab of the **Object Properties** dialog box, the formula will only be used for those nodes or elements that the object affects (see "Objects").

Formulas are, in essence, mathematical expressions that can be evaluated to produce real-numbers, integers, Boolean values, or text. A formula can include constants and any of the operators or functions described in the following two sections. A text constant must be enclosed in double quotes. Boolean constants must be either "True" or "False". Numeric values that do not have a decimal point and that are not expressed in engineering notation (such as 1E0) are considered integers. Other numeric values are real numbers. The period is always used as the decimal point in real numbers in formulas even on computers where the regional settings indicate that some other character is used as the decimal point. Integers are 32-bit values. Real numbers are double-precision values. Spaces, tabs, and line breaks in formulas are considered white space and are otherwise ignored. Loop constructs such as "while", "for", and "do" are not supported.

Importing MODFLOW Models

The **Import MODFLOW Model** dialog box is used to import an existing MODFLOW–2005 model into ModelMuse. It is displayed either by selecting **File|Import|Import MODFLOW–2005 Model...** in the Main Window or by selecting **Import MODFLOW–2005 model** in the **Start-Up** dialog box.

X Origin and **Y Origin** represent the corner of the grid at row 1, column 1 (the upper, left corner). **Grid angle** represents the angle in degrees that the grid is rotated counterclockwise. The **Name file** is the MODFLOW input file that lists the names of the other input and output files for MODFLOW.

The imported model will have a series of objects that define the properties and boundary conditions of the model. All of these objects will be hidden initially. Once imported, the user can edit or execute the model in the normal way.

Although, ModelMuse can import MODFLOW models, it has some intrinsic limitations. Boundary conditions, such as wells, will be imported as points at the centers of cells. The exact coordinates of the wells are not preserved in the MODFLOW input files. Similarly, rivers are imported as a series of points rather than linear features because they are represented as a series of cells in MODFLOW rather than as line segments.

Not all packages are imported into ModelMuse. The following packages or input files are imported: Discretization input file, Basic package, Zone arrays input file, Multiplier arrays input file, CHD package, RCH package, WEL package, DRN package, EVT package, GHB package, RIV package, DE4 package, GMG package, SIP package, and PCG package.

The way that data are specified in MODFLOW can limit the ways in which an imported model is modified. For instance, most of the boundary condition packages such as the Well, River, Drain, and General-Head Boundary packages, define the locations of the boundary conditions using layer, row, and column numbers. If the grid is changed after importing a model, these boundary conditions may no longer be located appropriately. When importing boundary conditions defined by lists of cells such as the Well, River, Drain, and General-Head Boundary packages, separate objects are imported for each stress period. If some of these boundary conditions need to be deleted or have their values changed, it may prove easier to delete the objects that define the boundary conditions entirely and to redefine them based on the original data than to edit them. Adding additional boundary condition objects, on the other hand, can be done with little difficulty. Properties that are defined by arrays in MODFLOW, such as hydraulic conductivity, are less of a problem because a value will be assigned to every cell. However, in some cases, it may be desirable to change the interpolation method used in certain imported data sets. The Nearest Point interpolation method is typically used in imported data sets.

The program **MF2005_Importer.exe** is used during the process of importing MODFLOW models. It is based on MODFLOW–2005. It reads data in the same way MODFLOW–2005 does and then writes that data in a simple format to the listing file for the model. ModelMuse then reads the data from the listing file. Thus, in the process of importing the model into ModelMuse, the existing listing file is replaced by one created by MF2005_Importer.exe. ***If the user wishes to preserve the existing output files from the model, a copy of the model should be imported rather than the original version of the model.***

Many MODFLOW–2000 models can be imported into ModelMuse in just the same way as the MODFLOW–2005 model. However, because of some differences in the input structure between MODFLOW–2000 and MODFLOW–2005, some MODFLOW–2000 models will have to be converted to MODFLOW–2005 models to be imported into ModelMuse. The program **MF2KtoMF05UC** can be used to convert MODFLOW–2000 to MODFLOW–2005 models. MF2KtoMF05UC can be obtained from *http://water.usgs.gov/nrp/gwsoftware/mf2ktomf05uc/mf2ktomf05uc.html*.

MODFLOW–96 models can not be imported into ModelMuse in the same way as MODFLOW–2005 models. However, the program **mf96to2k.exe,** which is distributed with MODFLOW–2000, can convert MODFLOW–96 models to MODFLOW–2000 models, which could then be imported as described above. MODFLOW–2000 can be obtained from *http://water.usgs.gov/nrp/gwsoftware/modflow2000/modflow2000.html*. In many cases, it will be necessary to provide the elevations of the tops and bottoms of the layers in order to convert MODFLOW-96 models to MODFLOW–2000 models because those elevations are not necessarily included in the MODFLOW input.

Executing the Model

To execute the model, select **File|Export|PHAST Input File** or **File|Export|MODFLOW Input Files** . A Save dialog box will appear in which the user is prompted to save either the PHAST transport data file or the MODFLOW name file. An **Execute model** checkbox will appear near the lower left corner of the dialog box. If it is checked, the model will be executed after the input files have been created. To execute a PHAST model, the user may need to create the PHAST chemistry data file too. ModelMuse does not create the PHAST chemistry data file. When executing a MODFLOW model, the listing file will be opened in Notepad when the model has finished executing.

Viewing Model Results

To view the spatial distribution of head, drawdown, or fluxes from a MODFLOW model, select **File|Import|Model Results...** Then select the file containing the results that are to be viewed. The **Select Model Results to Import** dialog box will appear. In it, select one or more of the results. The results will be imported into ModelMuse and can be used to color the grid just as can be done with any other data set. Spatial data from both MODFLOW and PHAST can be displayed in Model Viewer (Hsieh and Winston, 2002). Time-series plots of data from MODFLOW can be displayed with GW_Chart (Winston, 2000).

Troubleshooting Data Set Values

A data set or boundary condition will sometimes take on an unexpected or undesired value at a particular location. When this happens, ModelMuse provides the user with tools both to detect the problem and to understand why it has occurred. The following steps can be used to diagnose why a value is or is not assigned at a particular location.

1. First, the user can check that the column, row, and layer that are being examined are the intended ones. If the cursor is over the grid in the top view, the column and row at the cursor location will be displayed on the status bar. Placing the cursor over the **Selection Cube** will cause the selected layer to be displayed on the status bar. Similar information can be obtained from the front and side views.
2. Next the user can check what value is assigned to a cell and the explanation for how that value was assigned. To do this, the user first selects **Data|Color Grid...** to show the **Color Grid** dialog box and colors the grid with the data set or feature of interest. Then the user selects **Data|Show Grid Values...** to show the **Grid Value** dialog box. Both the value and the explanation are shown in the **Grid Value** dialog box.
3. If the value was assigned by the formula of the data set, the user can check the values of each of the other data sets referenced by the formula until one is found that has an unexpected value. If none of the referenced data sets has an unexpected value, the problem may be that the user does not understand exactly what the formula does. To understand the formula, the user can select **Data|Edit Data Sets...** to display the **Data Sets** dialog box. Next the user can select the data set of interest and click the **Edit Formula...** button to display the **Formula Editor**. The **Formula diagram** in the **Formula Editor** can be used to help understand how the formula works. If the formula makes use of any functions, the user can select those functions in the list of functions on the right and click the **Function help** button to display the documentation for the function.
4. If the data were assigned by interpolation, the problem may be that the user does not realize what **interpolation method** was used or how it works. To determine which **interpolation method** was used, the user can select **Data|Edit Data Sets...** to display the **Data Sets** dialog box. The **interpolation method** will be displayed. The user can then read the documentation for the selected **interpolation method**. In addition, the **Show or Hide Objects** dialog box can be used to show all

5. If the data were assigned by PHAST-style interpolation, the user can select **Data|Edit Data Sets...** to display the **Data Sets** dialog box. The interpolation direction, values, and distances or mixture formula. More information about PHAST-style interpolation is available in this report.
6. If the data were assigned by being enclosed or intersected by an object, the user can check if the object that was used to assign the value was the one that was intended. The user can also check exactly what it means to be enclosed or intersected by an object as documented on the help for the **Properties** tab of the **Object Properties** dialog box.
7. If the value was assigned by the wrong object, it could be because the value assigned by the desired object was overwritten by the value assigned by the other object. To check this, the user can first select the other object. The **Select Object by Name** dialog box may be helpful for this purpose. The user can then display the properties of the object that assigned the values to the data set in the **Object Properties** dialog box. The user can check the **Object order** for both the object that was intended to assign a value and the one that actually assigned the value. An object with a higher **Object order** can overwrite a value assigned by an object with a lower **Object order**. The Object order can be changed using the **Rearrange Objects** dialog box or by selecting an object, right clicking on the top, front, or side view, and choosing **To Front**, **To Back**, **Forward One,** or **Back One**.
8. If the value was assigned by the wrong object it could be because the desired object did not enclose or intersect the cell or element in the third dimension. The user can display the properties of the object in the **Object Properties** dialog box and check the number of third-dimension formulas and check that the formulas have the intended values. It can also be helpful to use the **3–D view** to check that the object encloses or intersects the desired cell or element. To do this, hide all the other objects and then depress the **Show 3D objects** button . Examining the other 2–D views can also be instructive. To do this, it is helpful to first set the color of the object in the **Object Properties** dialog box and then check the other views of the model to check that the object encloses or intersects the desired cell or element. The command **Grid|Select Column, Row, or Layer** can be helpful when checking a particular location. If the object does not enclose or intersect the desired cell or element in the third dimension and it is not clear why not, it can be helpful to check the position of the object in the third dimension. To do this, the user creates one or more 2–D data sets. The default formulae for these data sets should be set to the same formulae used to assign the third dimension of the object of interest. The grid can then be colored with the new data set and individual values of cells at locations of interest should be checked. In objects with formulas for a higher and a lower third-dimension coordinate, the user may find that the formula that is intended to give the higher value actually gives a lower value.
9. If the value of a boundary condition is being examined, the user can check the time for the boundary condition that is being used to color the grid in the **Color Grid** dialog box. Then the user can display the properties of the object in the **Object Properties** dialog box and check that the object assigns boundary conditions at the time that is being used to color the grid.
10. If a boundary condition is being examined, and no value is assigned to a cell or element by the desired object, check in the **Object Properties** dialog box that the object assigns values to either enclosed or intersected cells or elements.

11. For most boundary conditions, objects used to assign the boundary condition are normally 3–D objects. (The number of third-dimension formulas is one or two.) If a boundary condition is not being assigned properly, check that the object that should be assigning the boundary condition is a 3–D object. The PHAST river boundary is an exception to this rule.
12. In MODFLOW models, the user can check that the top of the cell is above the bottom of the cell.
13. In MODFLOW models, boundary conditions can not be assigned to cells in nonsimulated layers. The user can select **Model|MODFLOW Layer Groups...** to display the **MODFLOW Layer Groups** dialog box and check that the selected layer is part of a simulated layer group.
14. The user can select **View|Errors and Warnings** and see any errors or warnings that are related to the problem.

Example

An example model for MODFLOW is presented here. The ModelMuse help system has additional examples for both MODFLOW and PHAST.

Some users may choose to use this manual by starting with this section. Such users would benefit from reading the Basic Concepts section first.

The goals of this example are to:

1. Introduce the ModelMuse main window.
2. Introduce objects.
3. Introduce some of the important ModelMuse dialog boxes.
4. Illustrate how objects can be used to define non-uniform layer boundaries in MODFLOW models.
5. Introduce how to work with MODFLOW parameters in ModelMuse.
6. Introduce how to select MODFLOW packages in ModelMuse.
7. Introduce methods for importing data and background images.
8. Introduce methods for viewing model results.

The model in this example has already been created and is used to illustrate a variety of features in ModelMuse. Some of the features that the model illustrates are (1) spatially varying layer elevations, (2) the use of parameters, multiplier arrays, and zone arrays to define aquifer properties, (3) defining boundary conditions, (4) importing data and images, (5) running the model, and (6) viewing model results. The model has ten rows, ten columns, and nine layers. The model has several specified head boundaries and a well. The spatial distribution of the hydraulic conductivity in the Layer Property Flow (LPF) package is defined using a combination of parameters, multiplier arrays, and zone arrays.

To open the example model, start ModelMuse (by double clicking on ModelMuse.exe) and select “Open an existing model”. Select ExampleModel.gpt and the model will open. It should look similar to figure 21.

The ModelMuse main window shows four views of the model. The view on the upper left is a view from the top. The view on the upper right is a view from the right side. The view on the lower left is a view from the front. The view on the lower left is a three-dimensional (3–D) view of the model. The user can edit the model in the top, front, and side views. The model can be viewed but not edited in the 3–D view.

Three point objects (the small squares) are visible on the top view of the model. One of the squares is black, indicating that it is the selected object. If the **Select objects** button  in the toolbar is depressed, it is possible to double click on an object to display the **Object Properties** dialog box. The Object Properties dialog box is how the user defines what role each object plays in the model. Another important dialog box is the **Data Sets** dialog box. Data sets define things such as the hydraulic conductivity of the aquifers and the elevations of the layer boundaries. Some data sets are created by ModelMuse, but the user can create additional data sets for other purposes.

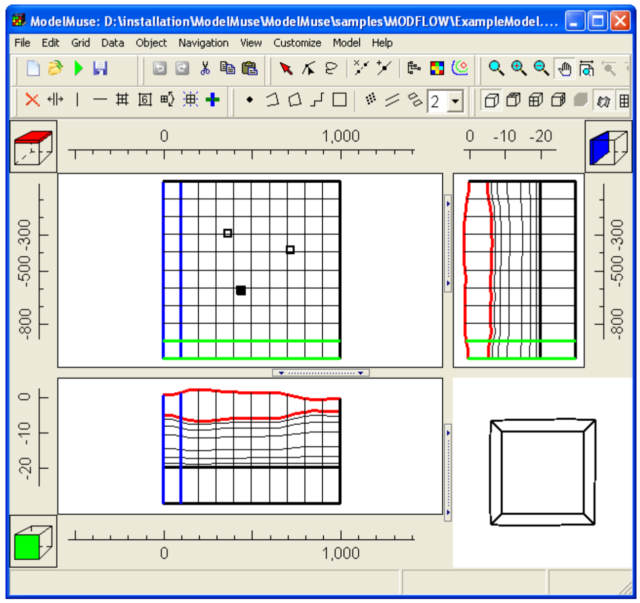


Figure 21. Initial appearance of ExampleModel.gpt.

Define Layer Groups

The example model has nine layers. To see how the layer boundaries are defined, select **Model|MODFLOW Layer Groups...** This brings up the **Layer Groups** dialog box (fig. 22). Although there are nine layers, only three aquifers are defined. This is because the middle aquifer is divided into seven parts. It is possible to see how the middle layer is divided by clicking on “Middle Aquifer” and selecting the “Discretization” tab. It shows that the layers grow progressively thicker toward the middle of the middle aquifer, and that a number of options exist for how a layer group such as “Middle Aquifer” can be discretized. Click **Cancel** to close the dialog box.

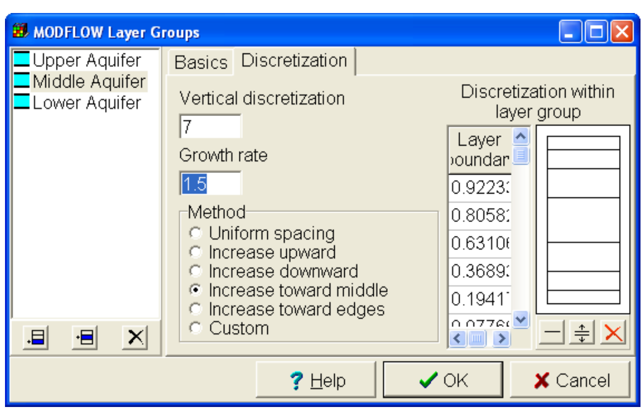


Figure 22. Layer Groups dialog box in ExampleModel.gpt.

Define Layer Boundaries

From the ModelMuse main window, first select **Object|Show or Hide Objects**  and expand “Data|Required|Layer Definition|Model_Top”. Note that four objects are used to assign values to Model_Top and that they do so by interpolation. Then select **Data|Edit Data Sets...** In the tree-view control on the left, select “Required|Layer Definition|Model_Top” (fig. 23). Note that it has a **Default formula** of zero and uses the **Inv. Dist. Sq.** (inverse distance squared) **Interpolation** method. Next select the “Upper_Aquifer_Bottom” data set. Note that its default formula is “-5. - Model_Top.” The middle and bottom aquifers have flat bottoms. However, the top and bottom of the top aquifer vary spatially. It is possible to start to see why here. The four objects define the elevation of the top of the upper aquifer at specific locations and interpolation is used to assign the top elevations based on those values. Because the top of the upper aquifer varies spatially and the default formula for the bottom of the upper aquifer includes a reference to the top of the upper aquifer, (-5 - Model_Top), it varies spatially too and because of the negative sign before Model_Top in the formula, the bottom of the layer is lowest where the top is highest. Click **Cancel** to close the **Data Sets** dialog box.

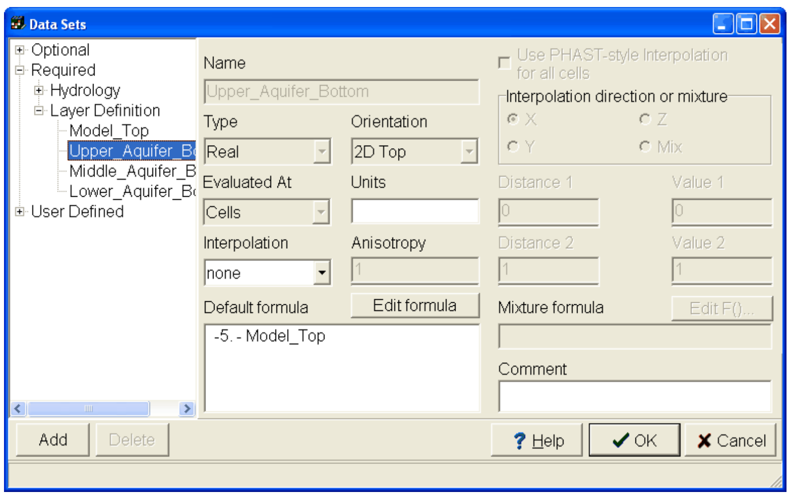


Figure 23. Data Sets dialog box in ExampleModel.gpt.

Using Objects To Define the Top of the Model

Select **Object|Show or Hide Objects...** The **Show or Hide Objects** dialog box (fig. 24) will be displayed. Uncheck the check box next to **All Objects** to hide all the objects. Then click on the “+” symbol next to “Data Sets” display groups of data sets that have objects that are used to set the values of those data sets. Similarly click on the “+” symbol next to **Required** and **Layer Definition**. Then check the check box next to “Model_Top” to display all the objects that are used to set the values of the “Model_Top” data set (fig. 25). On the top view of the model, there are now several objects that determine the elevation of the top of the model. The elevation is determined by interpolation among these objects. To get a better view of the top of the grid in the 3–D view (fig. 26*A*), select **View|Show Top Grid**. Then on the 3–D view of the model, click and drag to rotate the view of the model to get a better view of the grid. To see the bottom of the upper aquifer, locate the cube with the red square above and to the left of the top view of the model. Click on the cube below the red square to move the selected layer down by one (fig. 26*B*). The 3–D view will be updated to show the bottom of the upper aquifer.

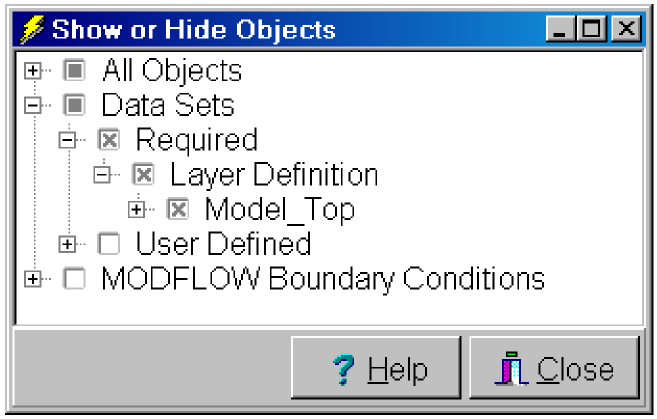


Figure 24. Show or Hide Objects dialog box in ExampleModel.gpt.

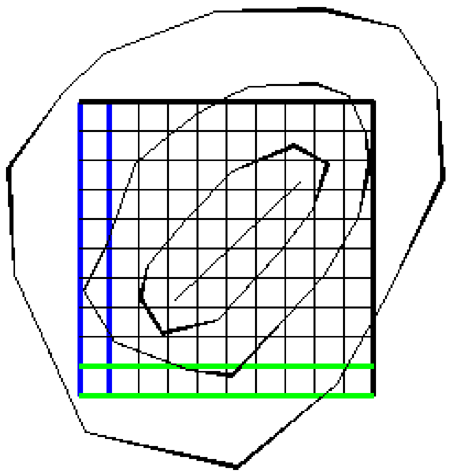
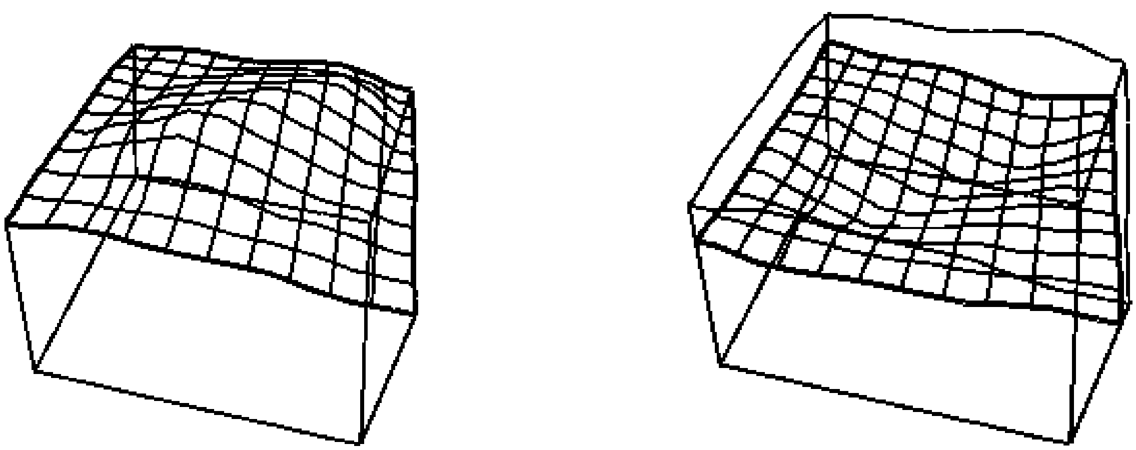


Figure 25. Objects used to define the top elevation in ExampleModel.gpt.



A *B*

Figure 26. Thee dimensional view of the grid (*A*) at the top of ExampleModel.gpt and (*B*) at the bottom of the first layer.

Defining Parameters

In the Layer Property Flow package, parameters can be used to define aquifer properties. In this model, two parameters are used to define the hydraulic conductivity. To see the definition of the parameters, select **Model|MODFLOW Packages...** On the left, select the LPF package, and then click on HK to see the parameter definitions (fig. 27). Parameter “HK_Par1” has a value of 150 and HK_Par2 has a value of 200. Both parameters use Zone arrays. HK_Par2 uses a multiplier array. Click **Cancel** to close the dialog box.

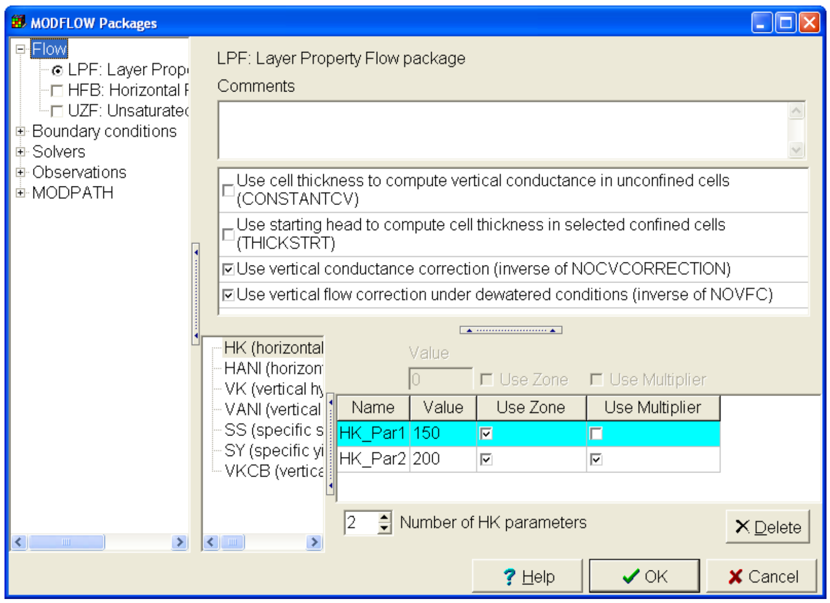


Figure 27. Parameters in the LPF package in ExampleModel.gpt.

View Zone Definition

To see the zone array for HK_Par1, select **Data|Color Grid...** to show the **Color Grid** dialog box, and then select “HK_Par1_Zone” from “Data Sets|Required|HK Parameter” and click **OK** (fig. 28). The blue zone is the area where HK_Par1 applies (fig. 29). However, by using the **Show and Hide Objects** dialog box (**Object|Show or Hide Objects**) to identify the objects used to define this zone, it can be seen that no such objects exist because no objects are listed for HK_Par1_Zone. To figure out what is happening, select **Data|Show Grid Values...** Move the mouse over the grid. When the mouse is over the blue cells, the “Value” is displayed as “True”. When the mouse is over the red cells, the “Value” is displayed as “False”. For both types of cells, the explanation that is shown is “set via default formula: ZoneID = 1” (fig .30). In this case, the value was set via the default formula for the data set. To see this default formula, select **Data|Edit Data Sets...** In this case the default formula has been set to “ZoneID = 1”. ZoneID is a user-created data set. In addition to HK_Par1_Zone, ZoneID has also been used in the default formula for HK_Par2_Zone whose default formula is “ZoneID <> 1”. Use the **Show and Hide Objects** dialog box to hide all the objects except the one that sets the value of ZoneID. The object is on the front view of the model (fig. 29B).

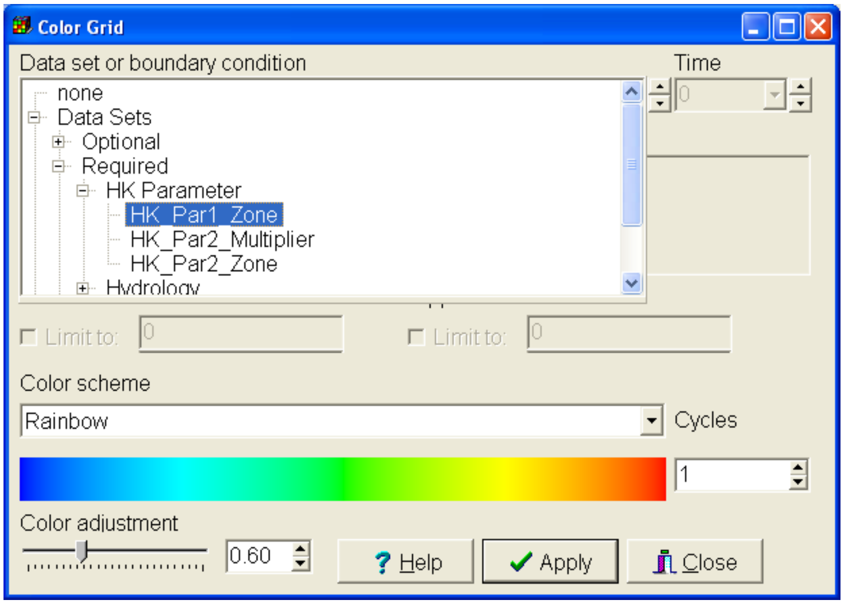
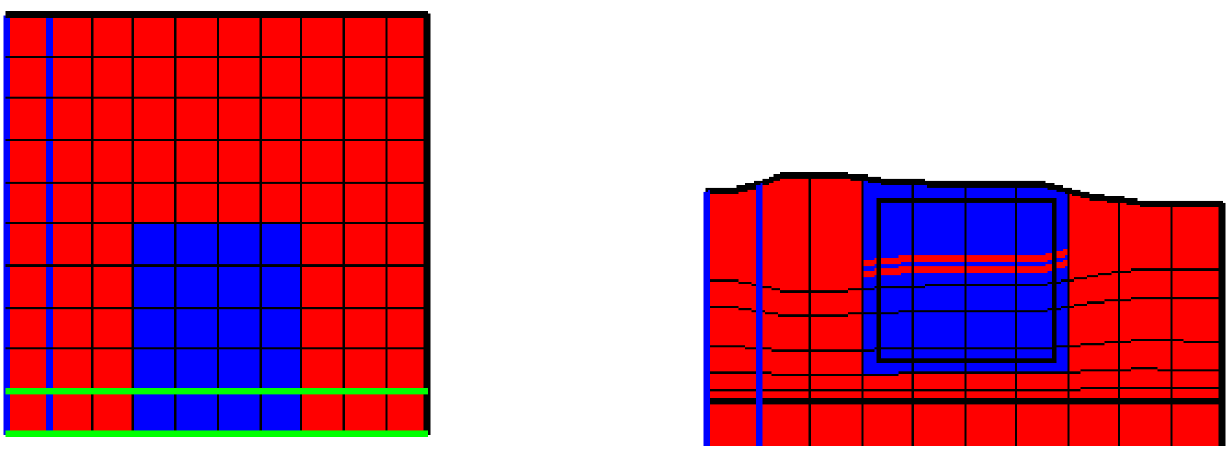


Figure 28. Selecting the HK_Par1_Zone data set in the Color Grid dialog box.



A *B*

Figure 29. The blue area represents the area where HK_Par1 applies—(A) Top view, and (B) Front view.

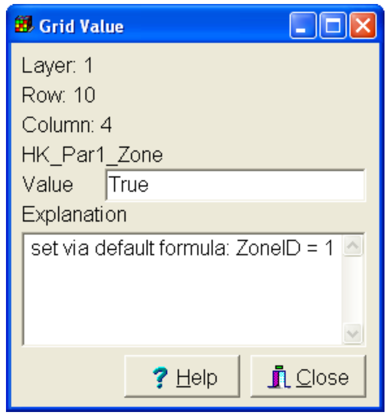


Figure 30. Grid Value dialog box.

ZoneID Object

To see how ZoneID is specified, first select **Data|Edit Data Sets...** and note that the default formula for ZoneID is “2.” Click **Cancel** to close the dialog box. To see how the “Zone_1” object sets the value for the ZoneID data set, select **Object|Select Objects**  and then double click on the rectangle object on the front view of the model. On the **Properties** tab of the **Object Properties** dialog box, note that the “Set values of enclosed cells” checkbox is checked (fig. 31). On the **Data Sets** tab, note that the check box next to “ZoneID” (under “User Defined”) is checked and its formula for this object is set to “1” (fig. 32). Back on the **Properties** tab, note that the formulas for the higher and lower Y-coordinate of the object are set to the front of the model and half-way to the back of the model. Thus, this object sets a value of 1 in ZoneID for all the cells inside the object for the front half of the model. Click **Cancel** to close the dialog box.

Because the elevations of the layers vary spatially, cells enclosed by the object are not the same in all the layers. Try moving the selected layer down (by clicking beneath the red square in the upper left) to see how the cells that are enclosed by the object vary from layer to layer. It can also be instructive to change the selected row by clicking behind the green square on the lower left **Selection Cube** to see why particular cells are inside the object.

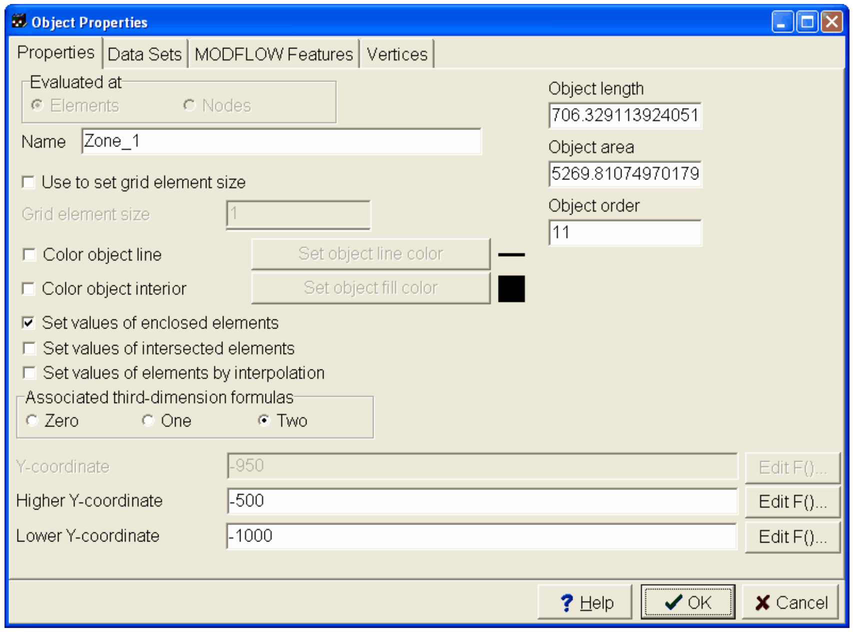


Figure 31. Properties tab of the **Object Properties** dialog box.

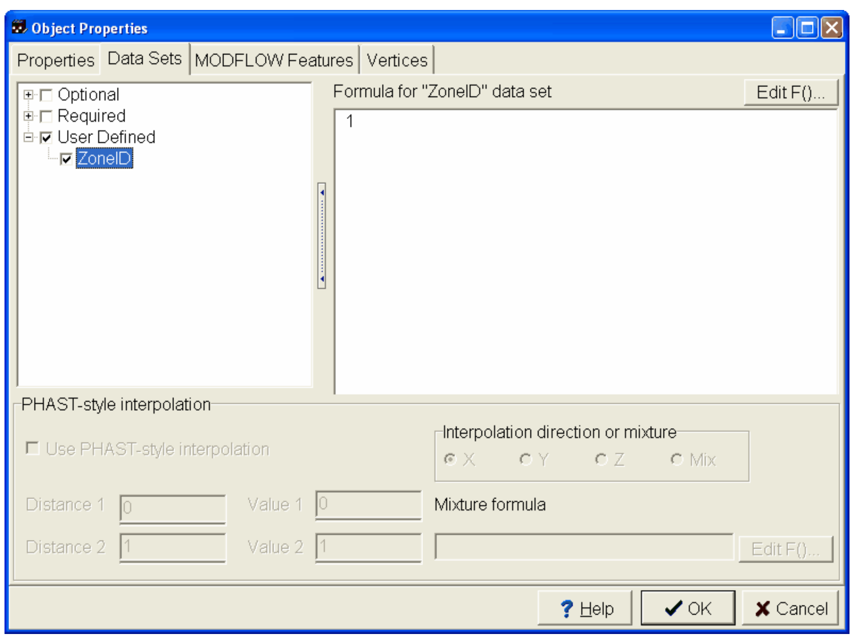


Figure 32. Data Sets tab of the **Object Properties** dialog box.

Multiplier Array

HK_Par2 uses a multiplier array. To see the distribution of this multiplier array, select **Data|Color Grid...** and select HK_Par2_Multiplier (fig. 33). These values are set by a default formula of "MyMultiplier". MyMultiplier is a user-created data set whose values are defined by three objects. Display those objects using the **Show and Hide Objects** dialog box. Double-click on one of these objects (the three black or hollow squares in figure 33) and see that the object sets the value of the data set via interpolation. By selecting **Data|Edit Data Sets...**, and selecting “User Defined|MyMultiplier” it is possible to see that the interpolation is **Triangle Interp**. Try changing the interpolation method to see how this affects the distribution of values in the multiplier array.

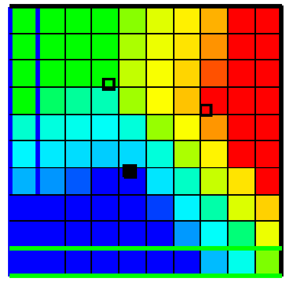


Figure 33. Cells colored with the multiplier array in ExampleModel.gpt.

Defining Stress Periods

To see how the MODFLOW stress periods are defined, select **Model|MODFLOW Time...** (fig. 34). One stress period is defined. The user defines the starting and ending times of the stress period, the length of the stress period, the desired length of the first time step, and the time-step multiplier. From these, the number of steps in the stress period is calculated. This is a bit different from the way the input for the stress periods in MODFLOW is normally defined; MODFLOW requires the number of steps in the stress period and calculates the length of the first time step. However, in ModelMuse, additional stress periods are inserted at times defined in the boundary conditions. Defining the stress period in this way makes it easier to insert them.

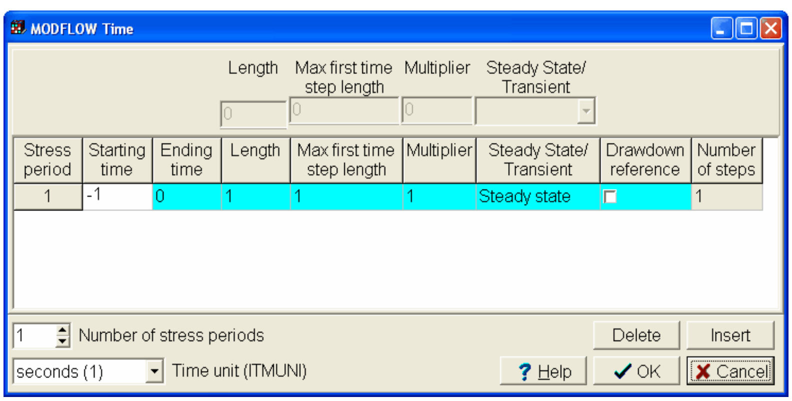


Figure 34. MODFLOW Time dialog box.

Selecting Packages

To use the Time-Variant Specified Head (CHD) package, it must be selected. To see that it is selected, select **Model|MODFLOW Packages...** and select “CHD” from the list on the left (fig. 35). Note that the checkbox for the CHD package is checked so that the CHD package will be used in the model. The locations of all CHD boundaries are defined by objects. Next we will see how those objects define the CHD boundaries. It is also possible to select PCG (under Solvers) to see how the values for the PCG package are defined.

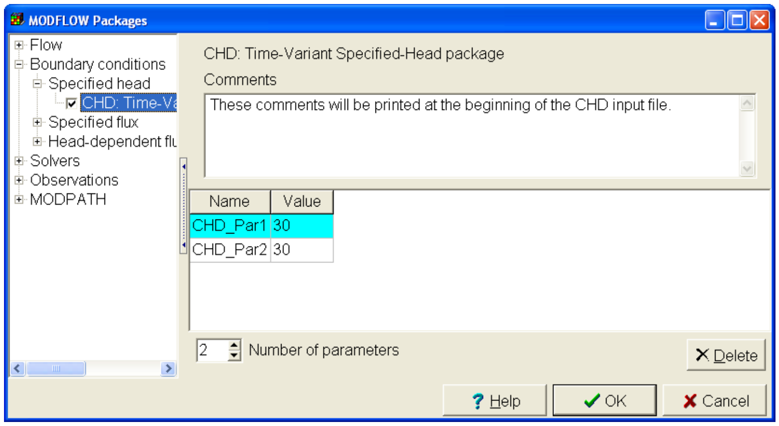


Figure 35. Parameter definition in the MODFLOW CHD package.

CHD Objects

Use the **Show and Hide Objects** dialog box to show the objects related to the CHD package (under "MODFLOW Features") and hide all the other objects. Three objects define the CHD Boundaries (fig. 36). Make sure **Object|Select Objects** is selected and click on the point object on the upper left. In the **Object Properties** dialog box, select the MODFLOW Features tab. The object uses the starting and ending times of the single stress period. (If there were more than one stress period, the times could encompass several stress periods.) The starting and ending heads were both set to 30. Two parameters are defined for the CHD package, but this object does not use either of them (fig. 37). Click on **Cancel** to close the **Object Properties** dialog box.

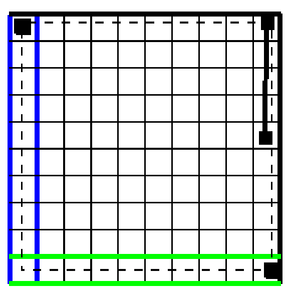


Figure 36. Objects defining CHD parameter boundaries in ExampleModel.gpt.

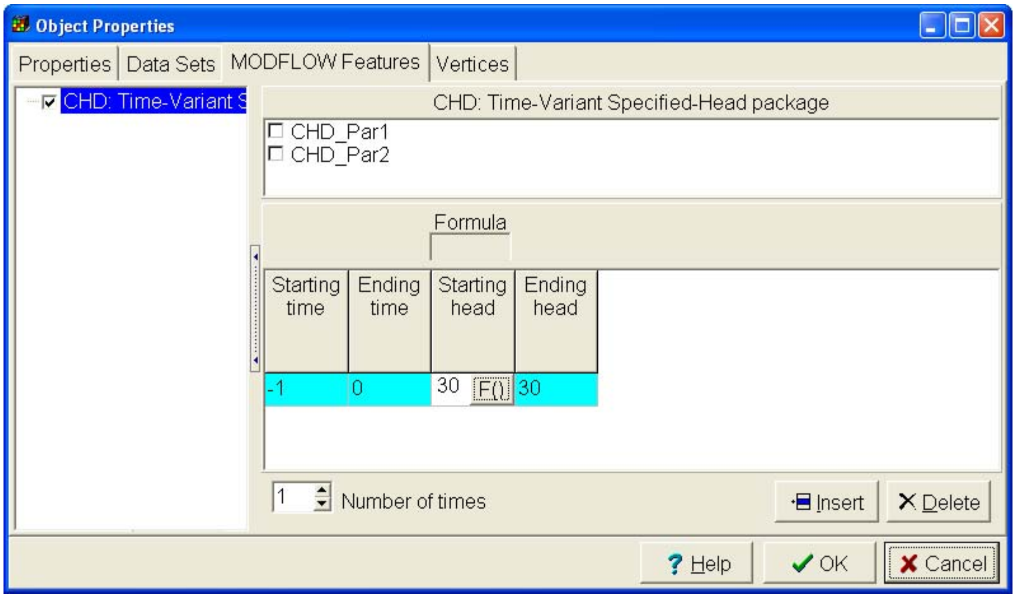


Figure 37. Properties of one of the CHD boundaries.

CHD Objects With Parameters

Click on the line object on the upper right. On the **MODFLOW Features** tab of the **Object Properties** dialog box, both the parameter check boxes are checked, which means that this object uses two parameters to define CHD boundaries (fig. 38). For each parameter, a formula is used to define a multiplier for the starting and ending head. In this case the formula for the multipliers CHD_Par1 for both the starting and ending heads is "Interpolate(Y, 1., -1000., 0., 0.)". The corresponding formulas for CHD_Par2 is "Interpolate(Y, 1., 0., 0., -1000.)". It is possible to edit the formulas directly in the cell in the table but for a complicated formula such as this, it is easier to use the **Formula Editor**. To view the **Formula Editor**, select one of the cells in the table containing the formulas, such as the one under "CHD_Par1 starting head multiplier". A button will appear in the table cell for the formula. Click it to display the formula editor (fig. 39).

The tree control at the top of the **Formula Editor** gives a diagram of the formula. It is possible to expand branches to see the arguments of a function. The tree control on the right allows the user to select items to insert into the formula.

Both formulas would interpolate between values of 1 and 0 depending on the value of the Y coordinate between 0 and -1000. However, the formula for CHD_Par1 would assign a value of 1 at -1000, whereas the formula for CHD_Par2 would assign a value of 0 at -1000 (fig. 40). The actual specified head assigned to the cells intersected by this object will be the sum of the values assigned by each parameter independently.

Click **Cancel** to close the **Object Properties** dialog box.

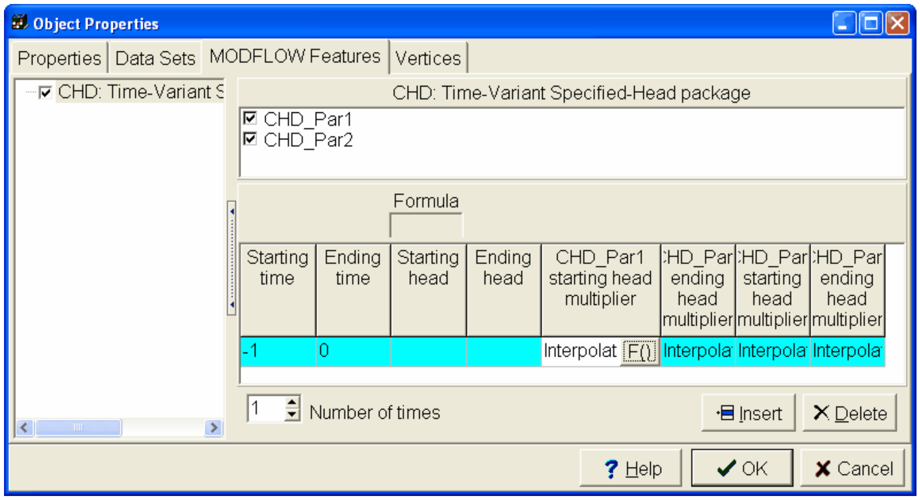


Figure 38. Object Properties dialog box showing the use of two parameters for one object.

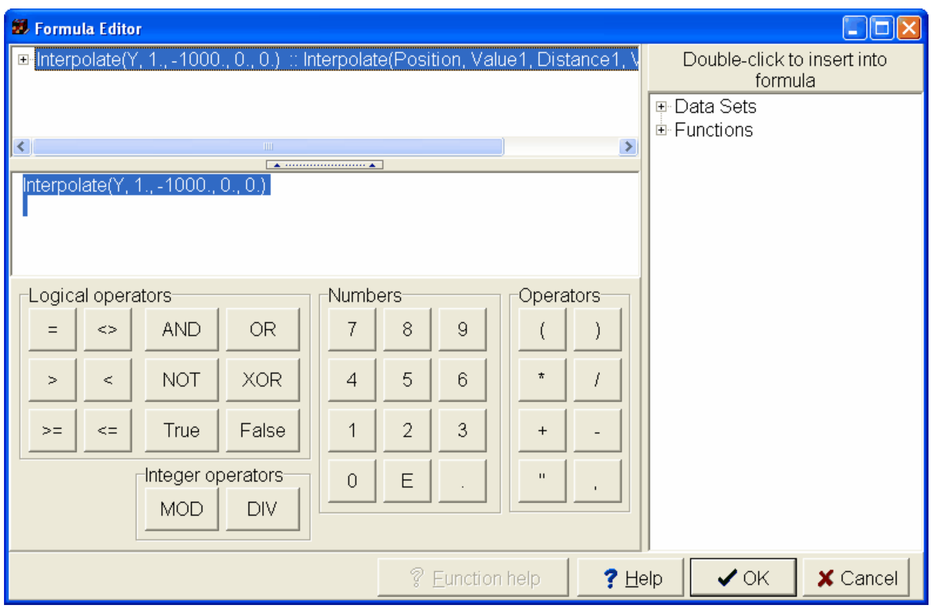


Figure 39. **Formula Editor** showing a formula used for interpolating the specified head of one of the parameters.

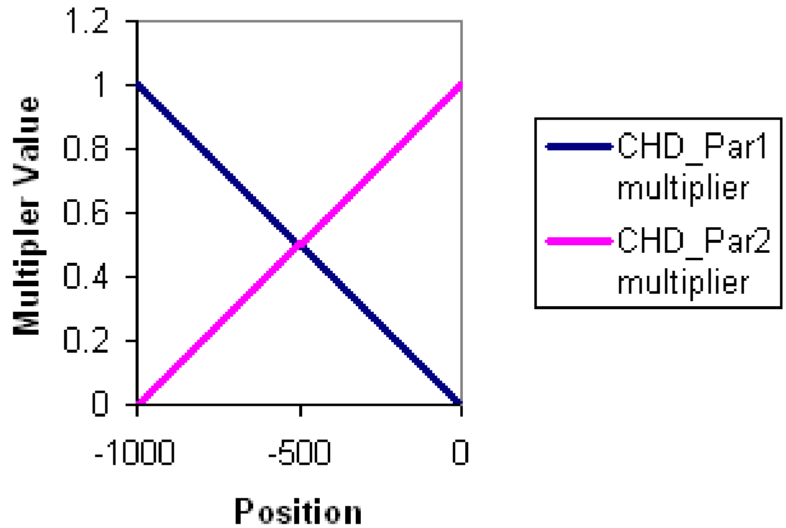
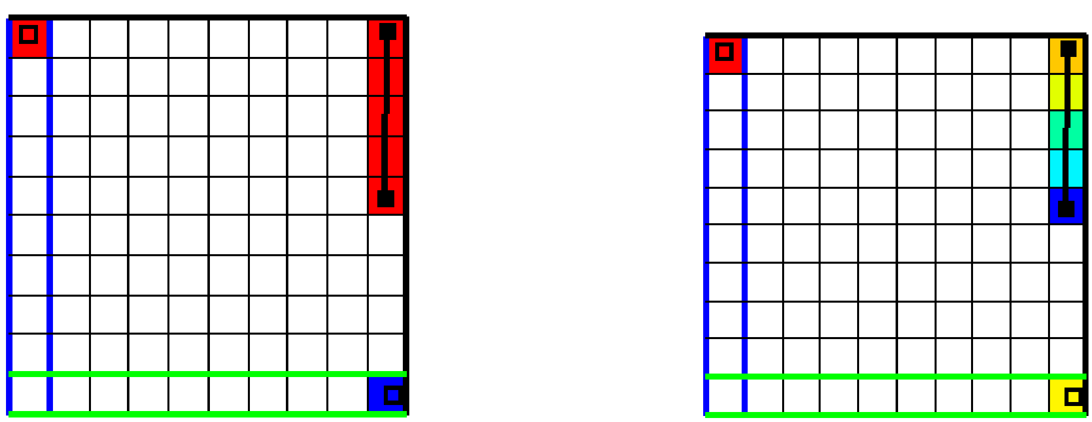


Figure 40. Graph of multiplier value as a function of position for the formulas for CHD_Par1 and CHD_Par2.

Displaying the Boundary Values

Next make sure the top layer of the grid is selected. (Use the red square on the upper left to select the top layer.) Then color the grid with the CHD Starting Head. (Select **Data|Color Grid...** and then select "Boundary Conditions, Observations, and Other Features|MODFLOW CHD|CHD Starting Head". The distribution of starting heads will be similar to that as in figure 41A below. Now change the value of the first CHD parameter from 30 to 20. The distribution will change so it looks like figure 41B. (Change the parameter value by first selecting **Model|MODFLOW Packages...** Then choose the CHD package under "Boundary Conditions|Specified Head". The CHD parameters will be listed. Change CHD_Par1 from 30 to 20.) Try using the **Grid Values** dialog box **Data|Show Grid Values...** and moving the cursor along the polyline object to see how the starting heads are assigned. The starting head for the line object in the upper right are assigned by combining the results of the two parameters that are used with that object.



A *B*

Figure 41. Starting head for CHD package (*A*) before and (*B*) after changing a parameter value.

Importing Shapefiles

ModelMuse can import Shapefiles (a common format for Geographic Information Systems; Environmental Systems Research Institute, Inc., 1998). Shapefiles can be obtained from many sources such as *http://nationalatlas.gov/maplayers.html*. For the purposes of this exercise, the Shapefile used is unimportant. To import a Shapefile, select **File|Import|Shapefile...** Then select a Shapefile to import. After reading the shapes, the **Import Shapefile** dialog box will appear (fig. 42). The user can decide which attributes of the Shapefile to import. If the coordinates of the shapes in the Shapefile are expressed in degrees, the user can convert them to Universe Transverse Mercator (UTM) coordinates using the Coordinate Conversion tab (fig. 43). If the user only wants to import some of the shapes, it is possible to restrict the shapes to import using the “Import criterion”. Any shape for which the import criterion evaluates to True will be imported. Any of the attributes of the shapes can be used in the import criterion.

After importing the shapes, it is possible that none of the shapes will be in the field of view. Use the menu item **Navigation|Go To...** to go to one of the imported shapes.

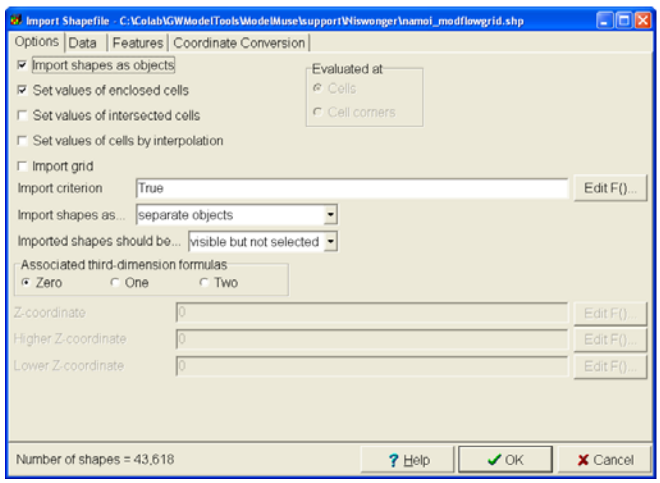


Figure 42. The **Import Shapefile** dialog box.

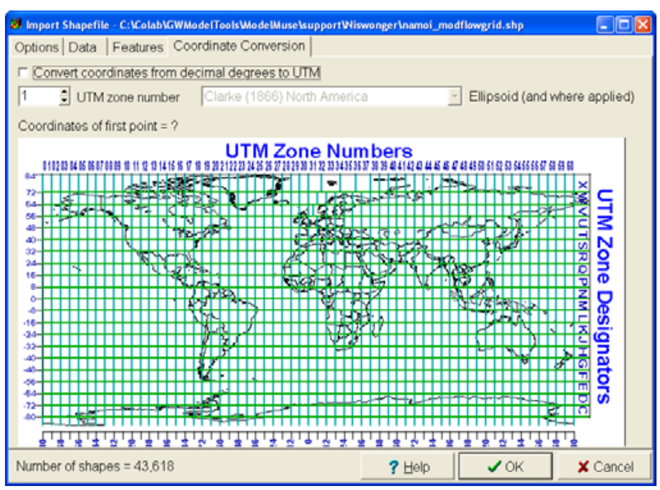


Figure 43. Coordinate conversion in the **Import Shapefile** dialog box.

Importing Images

To import an image, select **File|Import Image...** to display the **Import Image** dialog box (fig. 44). Then click the **Select Image** button and select the image to import. For the purposes of this exercise, the image used is unimportant. Identify several locations on the image with known real-world coordinates and click on them. A dialog box will appear in which to specify the real-world coordinates. When at least two sets of coordinates have been specified, the user can click the **OK** button to finish importing the bitmap. The image will be plotted at the specified location.

Figure 44. Import Image dialog box.

Importing Transient Data

One common task in modeling is specifying transient data. For example, there may be a well whose pumping rate changes from time to time. An example of such a spreadsheet, PumpingRates.xls, is distributed with ModelMuse. ModelMuse makes it easy to copy such data from a spreadsheet into the **Object Properties** dialog box. For example, with a spreadsheet such as the one in figure 45, the user could simply select the block of data to import and copy it to the clipboard. Then the user could select the upper left cell in the table of well pumping rates in the **Object Properties** dialog box (fig. 46) and press Ctrl-V on the clipboard to import the entire block of data into the model. If required, the number of rows in the table will automatically increase to accommodate the number of rows of data. It is not necessary to redefine the stress periods in the model. ModelMuse will determine the appropriate stress periods using all the transient data in the model. The example model has a single well (with a pumping rate of zero).

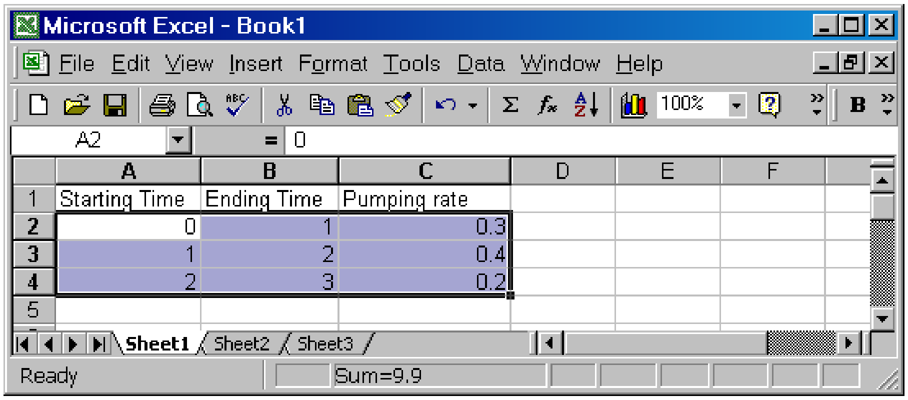


Figure 45. Spreadsheet containing transient data.

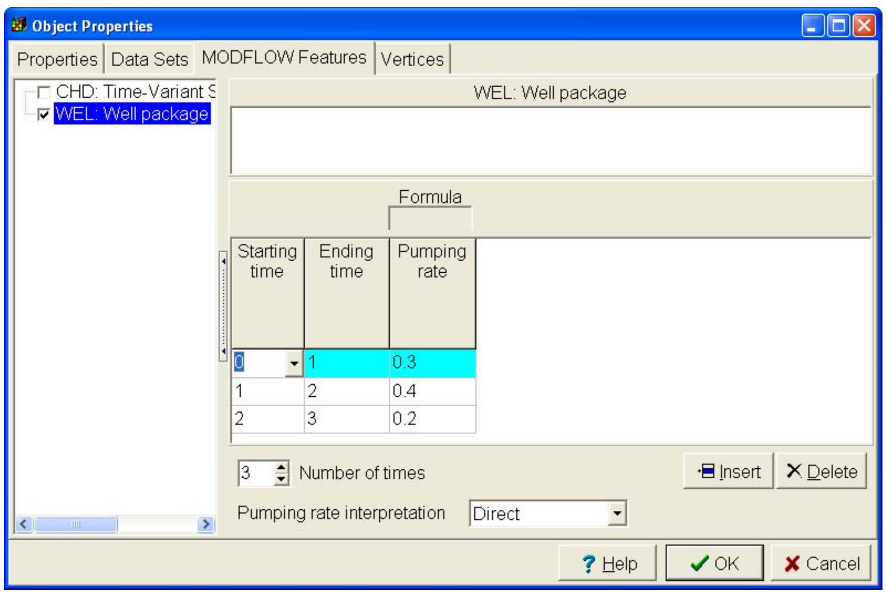


Figure 46. Transient data transferred to ModelMuse.

Global Variables

ModelMuse can create **Global Variables** (**Data|Edit Global Variables...**). Global variables can be used in formulas in ModelMuse in the same manner as data sets are used. Global variables play a role in ModelMuse similar to parameters in MODFLOW; they allow numerous model inputs to be changed at a single location.

One way that global variables could be used is as parameters in parameter-estimation. The “gpt” files created by ModelMuse are text files and the section containing the global variables is clearly marked. Thus, it is possible to create a UCODE template file that parameter-estimation program UCODE–2005 (Poeter and others, 2005) could use to vary the values of global variables. UCODE could then execute ModelMuse to create the model input files and execute the model. For modeling programs that do not have a built-in way of defining parameters, this could be useful.

There is a potential pitfall with using global variables as parameters for parameter estimation; there may be a nonlinear relationship between global variables and model inputs. For example, the user could define a global variable named “A” and then define hydraulic conductivity zones with inconsistent usages of A (in one zone the formula could be “A” and in another the formula could be “Sqr(A)”). In MODFLOW, there is always a linear relationship between parameters and model inputs. If additional nonlinearity is introduced, parameter-estimation may be difficult. The meaning of the parameters may also be problematic. In the example above, the physical meaning of A is unclear. This contrasts with MODFLOW in which, for example, HK parameters can be understood as representing hydraulic conductivity. Users would need to be careful how they use global variables if they wish to use them as parameters in parameter-estimation.

Executing the Model

To execute the model, select **File|Export|MODFLOW Input Files** and make sure the **Execute model** checkbox is checked. Select the location to save the name file and all the input files for the model; the model input files will be created in that location and the model will be executed. When the model input files have finished being exported, a dialog box showing any error or warning messages about the model will be displayed. A command-line window will open in which MODFLOW will execute. When MODFLOW is finished, the listing file from MODFLOW will open in Notepad.

Viewing Model Results

To view the heads, drawdowns, or fluxes calculated by the model, select “File|Import|Model Results...” and select the ExampleModel.fhd. The **Select Model Results to Import** dialog box will appear (fig. 47).

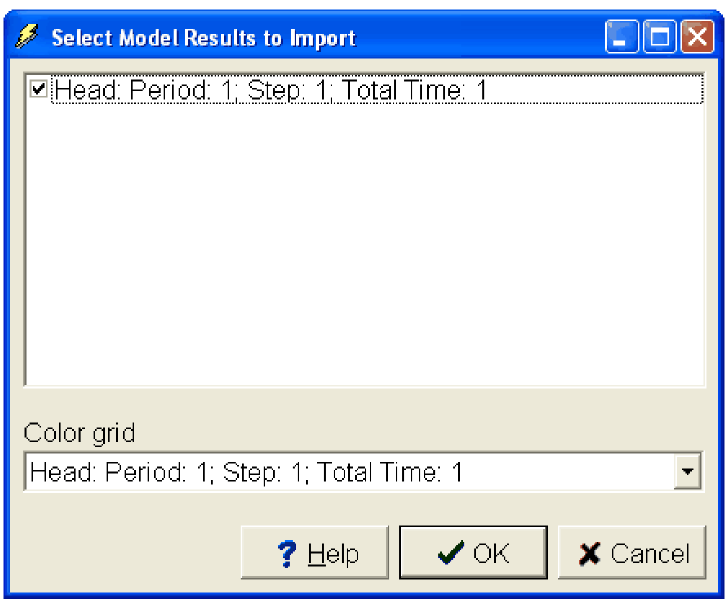


Figure 47. Select Model Results to Import dialog box.

In this dialog box, the user can select the data to import. In this case, the model only has one time step so there is only one set of data to import. It is also possible to color the grid with one of the data sets that is imported by selecting it in the **Color grid** combo box. The first, and in this case, only data set that is selected will be automatically selected in the Color grid combo box. The user can choose a different data set (if there is one) or “none” can be selected.

Click the **OK** button. The grid will then be colored as illustrated in figure 48.

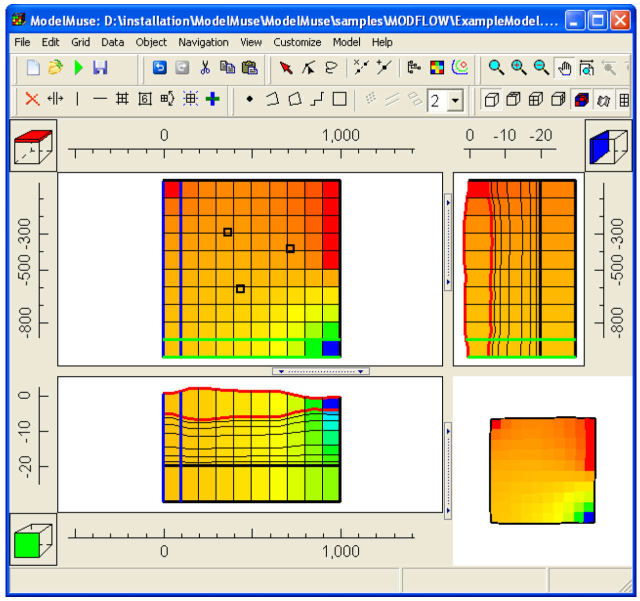


Figure 48. Simulated heads in ExampleModel.

In addition to heads, ModelMuse can also import drawdown and flux terms so that the user can plot them. To import them, the user merely needs to select the appropriate file to import. For time-series plots, a separate program, GW_Chart (Winston, 2000), can be used (*http://water.usgs.gov/nrp/gwsoftware/GW_Chart/GW_Chart.html*).

Limitations

The user should be aware of certain limitations that may make it difficult or impossible to use ModelMuse in certain circumstances. ModelMuse can use large amounts of memory—often more memory than is required by the modeling program to execute the model after the model input files are created. For some large models, the amount of memory required may be more than the computer has available; ModelMuse will fail if it runs out of memory. In addition, ModelMuse is not able to use more than three gigabytes of memory; if more memory than that is required, ModelMuse will fail.

To conserve memory, ModelMuse stores some data in temporary files. For large models, these files can be large. If the computer runs out of disk space on the drive containing the temporary files, ModelMuse will fail. When ModelMuse closes normally, the temporary files will be deleted. However, if ModelMuse closes abnormally, such as by having its process aborted in the Windows Task Manager, the temporary files will be left on the disk. When starting, ModelMuse will check for such orphaned files and will delete any it finds. The location of the temporary files will depend on the operating system used and how the operating system is set up. On Windows XP using the default configuration, the temporary files will be in a hidden directory named C:\Documents and Settings\<user name>\Local Settings\Temp\ModelMuse where “<user name>“ is the name under which the user is logged into the computer.

When working with large models, it may be necessary to use a computer with a large amount of memory and a large amount of free disk space on the drive in which temporary files are stored. As an experiment to see how much memory and disk space would be required for a large MODFLOW model, such a model was created in ModelMuse. The model had 3.36 x 10^6 cells (1,400 columns, 800 rows, 3 layers), and a single stress period. The CHD, WEL, RCH and EVT packages were active in this model. The maximum memory in use while the MODFLOW input files were exported was 1.2 GB. The temporary files created while the input files were exported totaled 55 MB. The time required to export all the input files was 21 minutes (processor speed 2.79 GHz, memory 2 GB).

Some operations can be time-consuming for large models. For example, drawing objects in the 3–D-view or drawing the cross sections of objects in a 2–D view other than the one in which the objects were originally drawn can be time-consuming for large models. For that reason, both these operations are turned off when the number of cells or elements in the grid exceeds 100,000. The user can turn them back on again by clicking the **Show 3D objects** button or selecting **View|Show 3D Objects**. The grid lines can be drawn in 2–D relatively quickly even in large models, but there may be so many of them that viewing the objects becomes difficult. The **Show 2D gridlines** button or **View|Show 2D Grid** menu item can be used to display or hide the grid lines in the model.

ModelMuse may not support all the options available for a model. To use unsupported options, the user may need to create additional input files or manually modify input files created by ModelMuse. (Manually created, additional MODFLOW input files can be included in the MODFLOW name file using the **Model|MODFLOW Name File...** menu item.)

Summary

ModelMuse provides a graphical user interface for creating the flow and transport input file for PHAST and the input files for MODFLOW. It uses objects (points, lines, and polygons) and formulas to define the spatial distribution of aquifer properties in a convenient fashion. ModelMuse can also be used to create the grids required by PHAST and MODFLOW and to edit nonspatial data required by PHAST and MODFLOW. The use of formulas has several benefits: (1) objects can have a third dimension that varies; (2) complicated distributions of values in data sets can be easily specified; and (3) changes to the grid can be made without requiring the user to reenter data. Just as the spatial distribution of data is independent of the grid, the temporal distribution of data in boundary conditions is independent of the temporal discretization of the model. ModelMuse combines all the times specified throughout the model to determine the proper temporal discretization of the model. ModelMuse allows the user to display the spatial input graphically. This display makes it easier for the user to avoid and detect errors in the input, thus making the modeling process more efficient and accurate. ModelMuse has built-in methods for importing data from DXF files and Shapefiles. It also has several built-in interpolation methods.

Acknowledgments

The author appreciates the thorough and thoughtful reviews of draft versions of this publication and software suggestions that were provided by Eileen Poeter of the Colorado School of Mines and the International Ground-Water Modeling Center and Arlen Harbaugh, Scientist Emeritus, USGS. Thanks also go to the beta testers and the members of the ModelMuse class at the U.S. Geological Survey 2008 Ground-Water Workshop for their feedback on ModelMuse. Iris Collies of USGS is thanked for careful and thorough editing. Anthony Winston, Jr. of Atlanta GA provided advice on improving the sound quality on the ModelMuse videos, C. Justin Mayers of USGS provided numerous helpful suggestions,

and Leonard Konikow and Paul Barlow both of USGS provided suggestions for improving the software and documentation.

References Cited

Environmental Systems Research Institute, Inc., 1998, ESRI Shapefile technical description: Environmental Systems Research Institute, Inc., Redlands, Calif., 28 p., accessed July 21, 2005, at *http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf*

Harbaugh, A.W., 2005, MODFLOW–2005, the U.S. Geological Survey modular ground-water model—The ground-water flow process: U.S. Geological Survey Techniques and Methods 6–A16, variously paged.

Hsieh, P.A., and Winston, R.B., 2002, User's guide to Model Viewer, a program for three-dimensional visualization of ground-water model results: U.S. Geological Survey Open-File Report 02–106, 18 p.

Parkhurst, D.L., Kipp, K.L., Engesgaard, Peter, and Charlton, S.R., 2004, PHAST—A program for simulating ground-water flow, solute transport, and multicomponent geochemical reactions: U.S. Geological Survey Techniques and Methods 6–A8, 154 p.

Poeter, E.P., Hill, M.C., Banta, E.R., Mehl, Steffen, and Christensen, Steen, 2005, UCODE_2005 and six other computer codes for universal sensitivity analysis, calibration, and uncertainty evaluation: U.S. Geological Survey Techniques and Methods 6–A11, 283 p.

Renka, R.J., 1996a, ALGORITHM 751. TRIPACK—Constrained two-dimensional Delaunay triangulation package: ACM Transactions on Mathematical Software, v. 22, no. 1, p. 1–8.

Renka, R.J., 1996b, ALGORITHM 752. SRFPACK—Software for scattered data fitting with a constrained surface under tension: ACM Transactions on Mathematical Software, v. 22, no. 1, p. 9–17.

Winston, R.B., 2000, Graphical user interface for MODFLOW, Version 4: U.S. Geological Survey Open-File Report 00–315, 27 p.

Winston, R.B., 2006, GoPhast—A graphical user interface for PHAST: U.S. Geological Survey Techniques and Methods 6–A20, 98 p.

Appendix 1—ModelMonitor

ModelMonitor is a program that monitors MODFLOW–2005 as it is running. It reads the listing file generated by MODFLOW and displays any error or warning messages. It also plots the percent discrepancy in the water budget. Based on this information the user can decide whether or not to allow a particular model to run to completion.

ModelMonitor has five tabs.

1. On the **Configuration** tab, the user selects the path for the MODFLOW executable and the name file of the model.
2. The **Monitor** tab displays the information that would normally be displayed on the screen by MODFLOW.
3. The **Listing** tab displays error and warning messages printed by MODFLOW in the Listing file. Error messages will be highlighted in red. Warning messages will be highlighted in yellow.
4. The **Results** tab displays a graph of the percent discrepancy in each stress period. If any of the percent discrepancies exceed one percent, that value will be plotted in red on the graph.
5. The **About** tab gives version information and contact information regarding ModelMonitor.

After the user has entered the path for the MODFLOW executable and the name file in the **Configuration** tab, pressing the **Start model** button will start the MODFLOW simulation.

It is possible to start ModelMonitor from the command line using the following syntax.

```
ModelMonitor.exe [[-m] <MODFLOW path> [-n][<Name file path>]]
```

Square brackets indicate optional arguments.
Angle brackets indicate that the text inside the brackets is to be replaced by the required information.
-m indicates that the following argument is the path for MODFLOW.
-n indicates that the following argument is the path for the name file.

If the optional arguments -m and -n are not used, the first argument is assumed to be the path for MODFLOW and the second is assumed to be the path for the name file. If any of the arguments contain spaces, the arguments must be enclosed by double quotes.

Example:

```
ModelMonitor.exe -m “C:\WRDAPP\MF2005.1_5\bin\mf2005.exe” -n “C:\WRDAPP\MF2005.1_5\
test-run\twri.nam”
```