WHEAT <=> GEM Translator Program GEM_WHT

Greg Pouch
Kansas Geological Survey

Abstract

Finite difference models are often used to model fluid flow, such as ModFlow and BOAST. Such models require two-dimensional arrays of input parameters for each model cell as well as input data and options that are specific to a particular program. The arrays of input parameters, such as permeability and withdrawal through wells, have geographic meaning and using a GIS – such as WHEAT – to manage the information and view the results reduces the difficulty of using such a model.

Traditionally, the input arrays were obtained from hand-edited text files, by using a dedicated pre-/post-processor, or by using a set of transfer programs specific to the combination of GIS and model. GEM improves on this situation by providing a common format that can be used by any GIS and by any finite-difference model.

This paper describes a MSWindows program that translates between GEM files and WHEAT, allowing WHEAT's EMAP programs to be used for manipulation of input and viewing of output from finite difference models.

Overview

Each finite difference model requires input in its own particular format. Preparing the input datasets and reading the output results can require as much effort as the actual modeling of the system. Such a situation significantly impedes productivity and hinders modeling.

Two solutions have been used to solve this problem: dedicated pre-processor and post-processor programs for specific models or model-families, such as Processing Modflow and Visual Modflow; and GIS-Model translator packages such as ArcMod. The pre-processor/post-processor approach has been more common on desktop systems and the GIS-Model link has been more common on mainframe/workstation systems.

The dedicated pre-processor/post-processor packages generally lack the ability to perform the spatial analysis needed to effectively model an area, such as assigning wells to model cells or using geologic maps to determine rock properties. The GIS-Model translator packages are generally complicated enough to make them non-portable between systems and difficult to use.

The GEM file standard used here solves this problem by breaking one large problem up into a small number of tractable pieces, which can be easily modified to take advantage of new models, new model features, and new GISs. It does this by storing model information in an easily-read Windows INI-style file.

This document describes the program GEM_WHT which translates between GEM files (INI files arranged according to the GEM specification) and WHEAT databases (Microsoft Access compatible MDB databases used with WHEAT).

WHEAT – the **W**indows-based **H**ydrogeologic **E**xploration and **A**ppraisal **T**oolkit – is a suite of user-friendly, public-domain MSWindows™ programs for natural resource exploration and management. The WHEAT programs use data stored in MSAccess®-format databases, so the data – geographic and attribute – can be manipulated easily with a wide range of programs, including MSAccess. Geographic features are stored in tables, and each feature is stored separately in a row. A set of features can have any number of attributes as columns. (The data can all be located in one table or spread across several tables and linked together to form a virtual table. There is a limit of 256 columns in any one table, but multiple tables can be used to store large numbers of columns.)

WHEAT was not intended to be the only software someone uses, but to be used in conjunction with other software such as Excel and Microsoft Access. The user of GEM_Wht is strongly advised that having both Microsoft Access and Excel, in addition to WHEAT, installed on her or his machine will make life much simpler. Very often, commercial software will be the only way or the better way to perform a task.

Organization of Data

Finite difference models require arrays of input parameters for each model cell as well as input data and options that are specific to a particular program. The arrays of input parameters, such as permeability and withdrawal through wells, have geographic

meaning and a GIS, such as WHEAT, is useful for managing the model-cell information and viewing the results.

Modeling consists of three phases: 1) defining the initial model, including its geography and number of cells and such; 2) an iterative phase of trial and error, where the modeller modifies values to attempt to match observations; and 3) using the model predictions to compare alternative courses of action.

Most effort in modeling is in the iterative process of trying values, comparing model results to measurements, and modifying the input values. The modeller repeats this sequence until satisfactory results are obtained. Sometimes, the new results are worse than the old ones, and reverting to the previous values is desired. Sometimes, it is useful to have several versions of the model input and output for other reasons, such as a best-case and a worst-case scenario. For these reasons, GEM_WHT is designed around the following assumptions:

1. A model grid is designed rarely and should be regarded as permanent
2. The parameters (permeability, storativity, and so forth) are modified many times
3. Multiple versions of input and output are needed.

To provide the most flexibility and ease of use, "models" in the database are divided into two parts: the model geography (one copy) and the properties (multiple copies corresponding to each revision). A property, such as permeability in layer 1, is stored as a section in the GEM file and as a field in the database.

Some parameters needed by a computer model, such as maximum number of iterations and convergence criteria, do not have spatial meaning and are not managed by the GIS, but by the computer-model-specific helper-program that translates between GEM files and computer-model input. Because running a finite difference model requires these two types of information, a GEM file contains information specific to a computer model and information derived from the GIS. Most activities are simpler if a pre-existing file that contains both sets of information is copied and modified, preserving previous choices.

Data requirements

IN THE WHEAT DATABASE

Finite difference models require two-dimensional arrays of property values, corresponding to the model-grid cells, along with a variety of other data that does not have spatial meaning, such as time steps and model options. The spatial aspects of the model-grid are usually unimportant to the finite difference model.

The model-cell geography is stored in WHEAT as a set of polygons in a table. Each polygon has a perimeter and a center point (X, Y) that can be used in spatial queries, as well as a ModelCellSerialNumber, a row number, and a column number. An entry for each model grid is kept in the table zWHEAT_ModelGrids. Do not ever modify any data

in a model grid or delete any rows (even if the model cell is unused) unless you are completely discarding the model.

Model input parameters and output parameters are stored separately in auxiliary tables. These tables are where you, the user, modify input values and view results. A query linking the model grid geography table with the cell values is created by GEM_WHT when you import data from a GEM file. Do not modify the ModelCellSerialNumbers. Information on model-cell property tables is kept in the table zWHEAT_ModelGridPropertyTables.

IN THE GEM FILE

A section [GEM_WHT] must be present in the GEM file and contain two entries indicating what database to use and what model grid to use. (In the geographic database, there can be many model grids, so both have to be specified.) There may be other entries giving historical information and such.

```
[GEM_WHT]
DatabaseName=e:\quivira\quivira.mdb
ModelGrid=Quivira One-Mile Cells
```

The rest of the file contains input or output data in the GEM File Specification, such as output heads or input formation thicknesses.

Directions

Before explaining particular tasks in GEM_WHT, it might be best to take a look at the overall modeling process and which programs do what. GEM_WHT handles the GEM file—WHEAT database translation. You need to supply a program, hereafter referred to as GEM_*Model*, that handles GEM—*Model* translation and provides additional functions such as determining the number of time steps. *Model* is some finite difference model, such as BOAST or MODFLOW.

The following list describes the modeling process using the GEM file scheme. Steps 4-9 are iterative and occupy most of the modeller's time.

0. Formulate a problem that can be solved with *Model*.

1. Create the model grid with GEM_WHT.
2. Create a GEM file from the model grid with GEM_WHT.
3. In the model translator GEM_*Model*, enter additional data into the GEM file. (At the end of this step, the GEM file should contain sections for data needed, instructions to the model-translator on how to use the sections to generate an input dataset, and other model-specific data. Make a copy of this file to use as a template GEM file.)

4. Run the model with GEM_*Model* by copying input from the GEM file into the model input file.

(Move model parameters from GEM into a *Model* input dataset using GEM_*Model* , run *Model*, then move the model results into the GEM file using GEM_*Model*.)

5. Import model results from GEM into the WHEAT database using GEM_WHT. (You may also want to contour the model results or perform other operations.)
6. View model results in EMAP programs
7. Modify the model parameters to get a better fit. This might involve using some combination of Access and WHEAT, or altering PropertyMappings in the GEM file.
8. Copy results from the database to a GEM file with GEM_WHT.
9. Repeat steps 4-8 until satisfied.

10. Make management decisions based on model results.

CONVENTIONS USED IN GEM_Wht

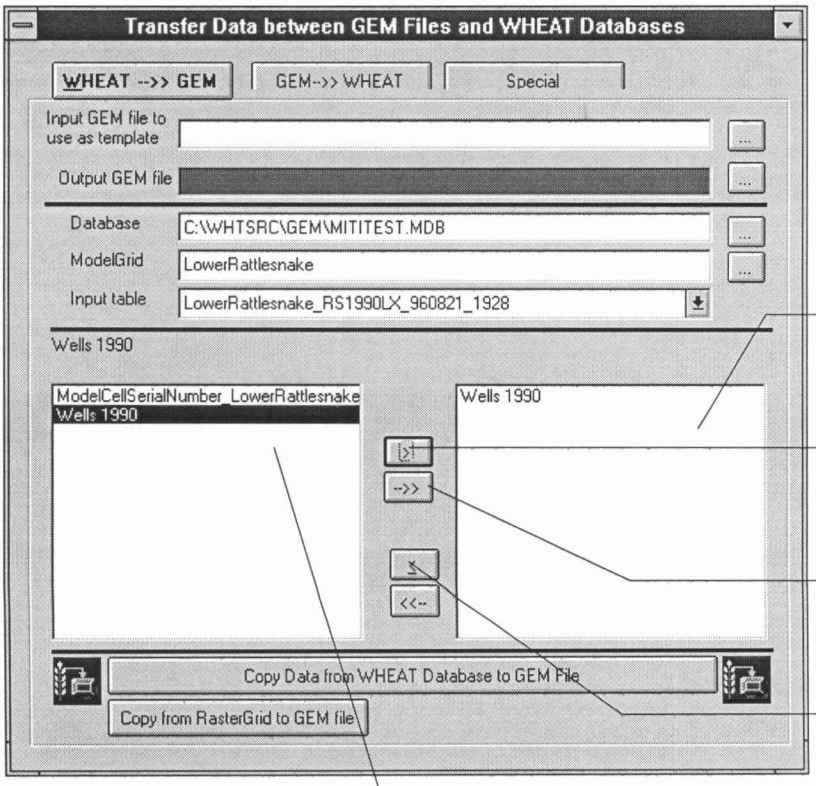
Running simulations is complicated. Transferring data between applications is complicated. Combining the two results in a very complicated package. Every effort was made to make this as simple and painless as possible, but this task is intrinsically complicated. Hopefully, the conventions used here will make it a bit more bearable.

Buttons labeled ... are "Browse" buttons. Clicking on a browse button brings up a dialog that will let you select a file or table for input or output. Often, browse buttons are located next to text boxes. In this case, you can type in the name of the file or table needed by hand, or look for it with the "Browse" command.

Sometimes it's hard to tell what's input and what's output. In these cases, the output is green (indicating new growth).

Many dialogs are structured like this one, where you are presented with a list of things available at left, and you add them to the list of things you have selected at right using the buttons.

Double-clicking in the left list adds it to the right list. Double-clicking on the right list removes it.



STARTING OUT

After deciding that you need to model and deciding such things as model location, model-cell spacing, and number of layers, the first step is to define the model grid. This means telling the computer where the model is located, how big the cells are, how many layers, and how the model-grid is oriented relative to the map coordinate system. This can be done in two ways: 1) determine grid coordinates elsewhere, import them into WHEAT, and make the appropriate entries in the GEM_WHT tables, or 2) design the model-grid entirely within GEM_WHT.

Create a New Model Grid

1. Choose the Special Page
2. Click on the button Create a new model grid. This will guide you through the subsequent steps.
3. Enter the X and Y coordinates of the center-point of the first cell
4. Enter the cell spacing (used for X and Y)
5. Enter the direction of the model's X-axis (bearing, degrees clockwise from north)
6. Enter the number of columns (X-dimension)
7. Enter the number of rows (Y-dimension)
8. Enter the number of layers
9. Enter a name for the model, name for area, and any comments.

The program will create the model grid and store it in the WHEAT database you selected. You should probably examine it in WHEAT and make sure this is the model area you wanted and that you like the layout of the cells.

Attach a Model Grid

This assumes you have already imported the model grid into WHEAT/Access into a table name *ModelName*_ModelGrid and have named a field ModelCellSerialNumber_*ModelName* and that you have columns in the table named Row and Column. ModelCellSerialNumber_*ModelName* should be the primary key, and Row and Column should be indexed.

1. Choose the Special Page
2. Click on the command Add a pre-existing model grid to the list. This will guide you through the subsequent steps.
3. Choose the name of the table (or possibly query).
4. Enter the number of columns (X-dimension)
5. Enter the number of rows (Y-dimension)
6. Enter the number of layers

CREATE A GEM FILE FROM SCRATCH

There are two ways to do this. The easier way is listed firs, but this requires that you already have model property tables in the database.

1. Select the WHEAT->GEM Page.
2. Open the database.
3. Select a model grid.
4. Select a properties table.
5. Choose the fields you wish to export.
6. Choose a file name to export the data into.
7. Click on the button labeled Copy data from WHEAT Database to GEM File.

1. Select the Special page
2. Click on the command Create a GEM file from scratch . This will guide you through the subsequent steps.
3. Choose the input database
4. Choose the model grid
5. Choose the output file name.

COPY MODEL-CELL DATA FROM A WHEAT DATABASE TO A GEM FILE

1. Click on the WHEAT-->>GEM button
2. Enter the name of the last revision GEM file in the textbox for Input GEM file to use as template. You can use the browse button next to it to find the file. This will open a database and select a model grid, as indicated in the labels near the property-tables list. If you don't already have GEM file based on this grid, create one from scratch as described in that section.
3. Enter the name of the output GEM file. (You can use the browse button to select a file.)
4. Select a property table from the list labeled Input table.
5. Choose the fields you wish to copy from the database to sections in the output GEM file.
6. Click on the command Copy data from WHEAT Database to GEM File

COPY MODEL-CELL DATA FROM A GEM FILE TO A WHEAT DATABASE

1. Click on the GEM-->>WHEAT button
2. Enter the name of the GEM file in the textbox for Input GEM file. You can use the browse button next to it to find the file. This will open a database and select a model grid, as indicated in the labels near the property-tables list.
3. Enter the name of the output table. You can accept the default if you don't mind the weird name.
4. Choose the sections you wish to copy from the GEM file to fields in the database with the arrow buttons.
5. Click on the command database Copy data from GEM file to WHEAT Database

You probably want to create a query so you can view this dataset using WHEAT, by clicking on the button labeled Make Query Linking This to Grid Cells , which will create a query named PropertyTableName QDF that contains the property cell values and the model geography.

MISCELLANEOUS TASKS

In addition to the main tasks, there are a number of sundry tasks like deleting old versions of the data and cleaning up databases that are needed. These are explained in this section. Many of these are not closely related to modeling, but are useful to keep the database in good shape or are useful otherwise.

Contouring

Neither GEM_WHT nor WHEAT EMAP programs contour data. You can contour model cell data by treating it as scattered XY data and using the WHEAT Contour program. If your model grid is oriented parallel to the coordinate axes, you can directly contour the grid by treating it as a raster grid.

Raster Grids

WHEAT uses gridded data internally for contouring. All such grids are stored in a table zWHEATtblRasterGrids, which contains the grids in binary format as a regular array of points. (This is not quite the same as model grids, because the cells must be oriented along coordinate axes and all cells must be the same size.) Data in these grids can be contoured using WHEAT's contouring program. You can copy data from and to this table using the commands Copy from RasterGrid to GEM File on the WHEAT->GEM page and Copy to RasterGrid in WHEAT database on the GEM->WHEAT page. Importing a raster grid requires information in addition to the gridded data, such as the DataType, the X_Origin and Y_Origin (location of first cell in the grid), DeltaX and DeltaY (cell sizes), BlankVal indicating what number represents unknown values, and an entry GridComesFrom giving historical information. It may be easier to export data from a raster grid, see what that looks like, and imitate that.

Deleting a table

You can delete a table and its entries in the ModelGridCatalog by choosing the Special page and clicking on Delete a model-cell data table . Be careful, because once it's deleted, it's gone forever.

Attaching data

If you copy a model-cell properties table or create one, GEM_WHT has no way of knowing of this until you manually update the appropriate links with the command Attach cell data to model-grid on the Special page.

If you generate a model-grid elsewhere and import it into Access, you need to list in the ModelGridCatalog by choosing Add a pre-existing model grid to the list on the Special page.

Blank Cell Property Table

You can create a blank model-cell property table by choosing Create a model-cell property table on the Special page.

Database cleaning

The space used by deleted records and tables is not reclaimed by Access in the normal course of use. You can use Access to clean the database by Repairing and Compacting the database, or you can follow the riskier course of choosing Clean up a database on the Special page.

Kansas Geological Survey
Open-file Report

Disclaimer

The Kansas Geological Survey does not guarantee this document to be free from errors or inaccuracies and disclaims any responsibility or liability for interpretations based on data used in the production of this document or decisions based thereon. This report is intended to make results of research available at the earliest possible date, but is not intended to constitute final or formal publication.