A Data Input Program (MFI) for the U.S. Geological Survey Modular Finite-Difference Ground-Water Flow Model

by Arlen W. Harbaugh

U.S. GEOLOGICAL SURVEY
Open File Report 94-468

Reston, Virginia
1994

U.S. DEPARTMENT OF THE INTERIOR
BRUCE BABBITT, *Secretary*

U.S. GEOLOGICAL SURVEY
Gordon P. Eaton, *Director*

For additional information write to:

Office of Ground Water
U.S. Geological Survey, WRD
411 National Center
Reston, VA 22092
(703) 648-5001

Copies of this report can be purchased from:

USGS ESIC -- Open-File Report Section
Box 25286, MS 517
Denver Federal Center
Denver, CO 80225
(303) 236-7476

PREFACE

This report presents a data input computer program (MFI) for the U.S. Geological Survey modular finite-difference ground-water flow model, commonly known as MODFLOW. The program has been tested by using it to enter data for a variety of model simulations, but it is possible that other applications could reveal errors. Users are requested to notify the U.S. Geological Survey if errors are found in this report or the program. Copies of the source computer program on diskette are available for the cost of processing from:

U.S. Geological Survey
NWIS Program Office
437 National Center
Reston, VA 22092
Telephone: (703) 648-6978

In addition to the MFI source, the ANNIE Interactive Development Environment (AIDE) software is also required in order to make an operational program. For Data General workstations and Sun Sparcstations, AIDE source code can be obtained from:

U.S. Geological Survey
Hydrologic Analysis Support Section
415 National Center
Reston, VA 22092

For IBM compatible personal computers, AIDE source code can be obtained from:

U.S. Environmental Protection Agency
Center for Exposure Assessment Modeling
960 College Station Road
Athens, GA 30613

The source program must be compiled using a Fortran compiler in order to make it run on a particular computer. Modifications to the program may be required in order to make the program work on computers other than the kind on which it was developed, which is the Data General Unix workstation.

CONTENTS

ILLUSTRATIONS

Page

TABLES

Page

A Data Input Program (MFI) for the U.S. Geological Survey Modular Finite-Difference Ground-Water Flow Model

by Arlen W. Harbaugh

ABSTRACT

MFI is a data input (entry) program for the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model commonly referred to as MODFLOW. Data for MODPATH, a particle tracking program for use with MODFLOW, can also be entered using MFI. MFI is designed to be easy to use and portable. It can also be used in conjunction with other programs so that the different parts of a model dataset can be entered using the most suitable program. Data are entered in MFI interactively through a series of display screens. MFI interfaces to a commercial spreadsheet program for entering two-dimensional arrays and lists of stress data. This report provides instructions for using MFI and information about making MFI run on different computers.

INTRODUCTION

Computer models of ground-water flow require input data in order to define the system being simulated. Most model programs read these data from one or more files rather than prompting for the data interactively. The reason for this is partly evolutionary. When ground-water models were first developed in the 1960's, the primary focus was on developing the numerical methods. The major challenge was being able to solve practical, real-world problems within the constraints of computer speed and memory. Computers were too expensive to use for data entry. Also, the simulations were generally small by today's standards, so it was relatively easy to prepare the data files using a card punch or text editor.

Today's computers can easily handle large modeling problems for which data preparation is a burdensome task. Considerable benefit can be obtained from an interactive program that helps prepare model input data. Such a program can prompt for the required data, check for errors, and ensure that files are created with the proper structure. Although it would be possible to add interactive data entry into the model program, this is not desirable because it would unnecessarily increase the required memory and increase the complexity of the program. Further, the wide variation in user preferences and needs makes it unlikely that any single method of interactive data input will meet all needs. It is better to have data entry separate from the model so that different data entry programs can be developed independently of the model program.

This report describes a data input program, called MFI, for the U.S. Geological Survey (USGS) modular three-dimensional finite-difference ground-water flow model (McDonald and Harbaugh, 1988), which is commonly called MODFLOW. MFI can also prepare data for MODPATH, which is a particle tracking program that uses flows calculated by MODFLOW (Pollock, 1994). Much of the data used by MODPATH is the same as that used by MODFLOW, so considerable redundancy is avoided by having one program prepare data for both.

Design Goals

The primary goals for MFI are:

1. Ease of use -- This goal is subjective to a large extent. Some of the desired attributes are: the user is guided in a logical manner through the data entry process, checking is done to make sure valid or reasonable data are entered, and help information can be displayed to avoid frequent referral to model documentation.

2. Ability to run on many kinds of computers -- This is an important requirement because USGS modelers use many kinds of computers, and the specific computers being used change frequently.

3. It must be possible to use other programs along with MFI -- Modelers use a variety of computer applications to help manipulate their data. The applications used vary widely depending on the specific situation such as the availability of specific computer software and the source of the data. For example in the USGS, many modelers use Geographic Information System (GIS) software to store and manipulate data. It is important to be able to use data from the GIS along with data entered using MFI.

4. Low development cost -- It would be easy to spend more on the development of a data editing program than was spent developing MODFLOW; however, considerably less resources were available for this effort.

5. Useful for a wide range of modeling problems from simple training exercises to large modeling projects.

Design Decisions

The following design decisions were made in order to fulfill the goals:

1. The programming language will be Fortran -- Many other languages would have been adequate, but the author is most familiar with Fortran. Also, MODFLOW is written in Fortran, so this makes it easy to read and write data for MODFLOW.

2. The ANNIE Interactive Development Environment (AIDE) will be used as the tool for providing user interaction in MFI (Kittle and others, 1989) -- The version of AIDE used in MFI includes enhancements that were added to AIDE after its original publication (P. R. Hummel, written commun., 1994). AIDE provides a set of tools for creating programs that interact with users through menus and "fill in the blanks" screens. AIDE is a character based system as compared to a graphical system. It is not as powerful as many of today's graphical systems, but it is portable. AIDE incorporates a help system and range checking of data. AIDE is sufficient for controlling the selection of options and entering miscellaneous data. It is insufficient for entry of large arrays.

3. Large arrays of data and lists of point stress data will be entered through the use of a commercial spreadsheet program -- Regardless of what interactive interface tool is used, writing a sophisticated user interface for entering arrays and lists of stresses would be too expensive and time consuming. By interfacing with an existing program, the greatest functionality can be obtained at the lowest cost. Further, it is likely that other commercial programs could be useful for generating arrays depending on user preferences and the specific modeling task. Thus, MFI was designed under the assumption that interfaces to other programs could be added.

4. MFI will primarily deal with gridded data -- That is, it is assumed that the required data values are known at all points in the grid. Although there is no intent to directly incorporate gridding capabilities within MFI, these capabilities could be obtained by using other programs along with MFI.

5. As a further method of allowing the user to maximize the use of other software, MFI will allow external files to be specified as part of a MODFLOW dataset -- These external files will not be read or modified by MFI, but the dataset will be setup so that MODFLOW will read these files. This is possible because MODFLOW allows arrays to be read from external files.

6. To avoid duplicate data, MFI will directly read and write MODFLOW datasets rather than maintaining a separate database.

USER INSTRUCTIONS

MFI works by displaying screens to which the user responds. Each screen includes a data window, an instruction window, a line of available commands, and an optional third window. The data window is used to display information, allow information to be entered (or edited), or allow a choice to be selected from a menu. The instruction window shows error messages and instructions about using screen commands. The optional window can be a help window, data limits window, or a scratch pad window in which the user can enter notes. Figure 1 illustrates the structure of a screen.

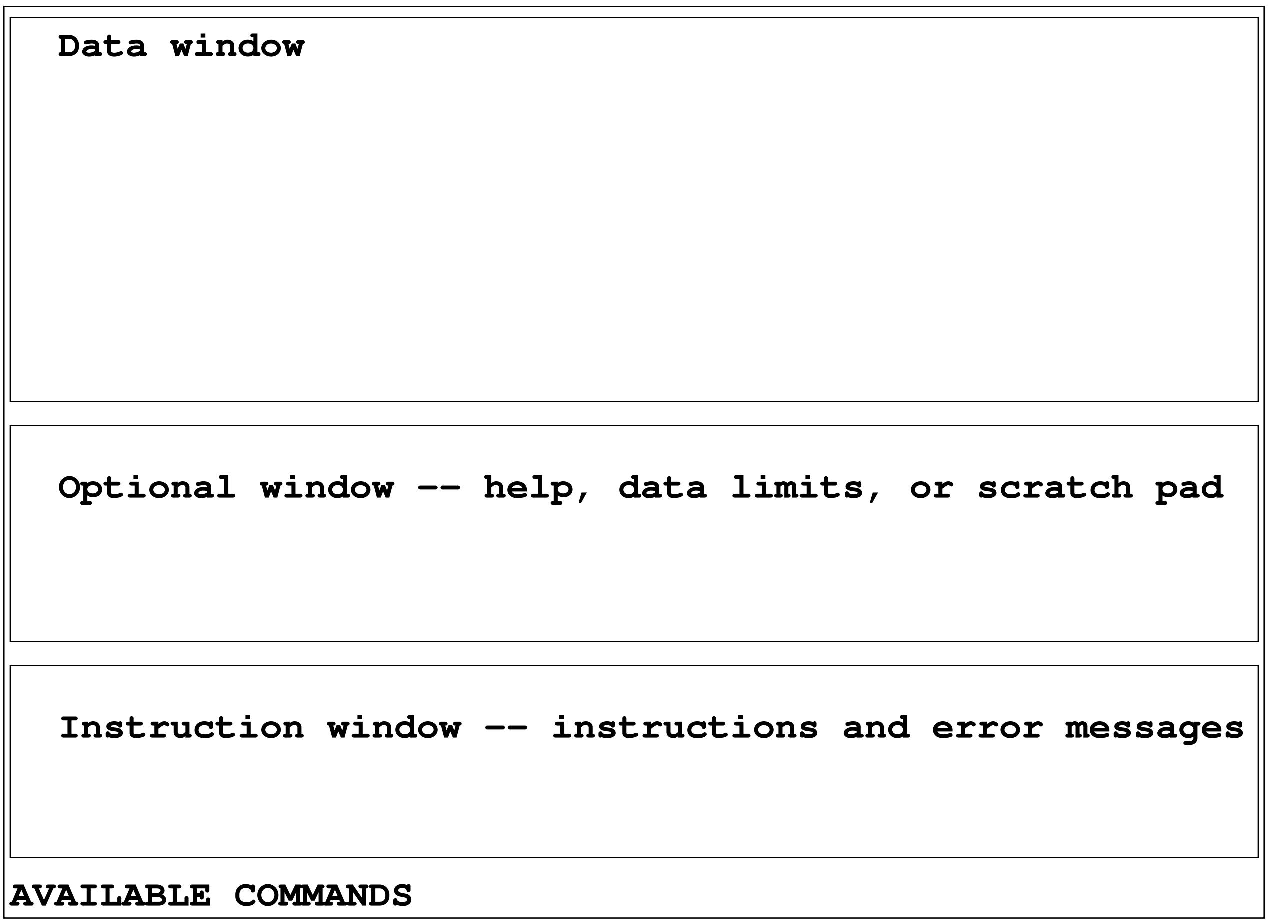


Figure 1. -- Screen structure of MFI.

The user enters data, makes a choice, or reads information displayed in a screen's data window, and then "accepts" the screen in order to proceed to the next screen. A screen is accepted by the F2 command described below. Choices are made in menu screens by using the arrow keys to move the cursor to the desired choice prior to accepting the screen. Data are entered by moving the cursor to desired fields and entering values.

Screen Commands

The commands available to a particular screen can vary. The available commands are listed at the bottom of each screen. Table 1 is a list of all the commands used in MFI screens.

Running MFI

MFI is started by entering

mfi

or

mfi *dataset*

where *dataset* is a dataset name.

If the dataset name is not entered as part of the command, then the first screen to appear will prompt for a dataset name. A dataset name consists of up to 8 characters. This name is used as the beginning part of the names of many of the files that will be created. These file names are constructed using the dataset name followed by a dot plus a 2- or 3-character suffix. Some examples are:

.nam is the suffix for the file used to keep track of all files used by MODFLOW;
.log is the suffix for a log file of information generated by MFI;
.bas is the suffix for the Basic Package file of MODFLOW;
.bcf is the suffix for the BCF Package file of MODFLOW;
.mpn is the suffix for the MODPATH name file;
.mp is the suffix for the main MODPATH data file.

The dataset name cannot include characters that are invalid as part of a file name on the particular computer being used.

If a dataset exists, it is read. If a dataset is new, the next screen prompts for confirmation that it is acceptable to create a new dataset. If so, the following screen prompts for the grid size: the number of layers, rows, and columns. These grid size values cannot be changed within MFI after this screen is accepted, so be careful to specify them properly. If it is not acceptable to create a new dataset, MFI stops without doing anything more. MFI stops at this point because it is assumed that the user intended to enter the name of an existing dataset, but may have forgotten the name. The user can then take whatever steps are necessary to determine the correct name and restart MFI.

Table 1. -- Screen commands for MFI.

Key	Command	Command Description
F1	Help	Creates an optional window that displays help information related to the data window. For screens in which data are being entered, help information is usually provided for each individual field. If a message is longer than the help window can show, the lines of text can be scrolled. When scrolling in the help window, processing in the data window is interrupted in order to allow the cursor movement keys to be used for the scrolling. To continue in the data window, press the F3 key. Scrolling in the help window is activated automatically when the help window is initiated if the current message is a long message. If the help window is active and a long message is encountered after a previous short message, a second entry of F1 will activate scrolling.
F2	Accept	Accepts the contents of the screen. This tells the program to capture any data entered in the screen and progress to the next screen. A menu screen can also be accepted by entering a carriage return or by typing enough letters of the desired choice to uniquely define that choice. For most screens in which data are being entered, screen acceptance occurs when a carriage return is used to terminate the value for the last field on the screen.
F3 or ;		The primary use for F3 is to terminate other commands. This use is described as part of the appropriate command descriptions (for example, F1). F3 is also a command by itself. When the F3 key is entered as a separate command, the display cursor moves to the command line; a command can then be selected by using the cursor keys to move the cursor to the desired command and entering carriage return. This is the only way to access the Oops command. Other commands are more easily invoked by using their own function keys, so using the F3 command is not generally useful for specifying commands other than Oops. The F3 command is available in all screens although it is never displayed as an available command. To exit this command, simply select one of the commands or enter the F3 command a second time.
F4	Prev	Go back to the previous screen. This command is included in screens for which the accept command (F2) results in more screens that eventually return back to the same screen. Without the F4 command, there would be no way to stop the process. For example, when entering hydraulic arrays for the Block-Centered-Flow (BCF) Package of MODFLOW, a screen first prompts for a layer number. After this, the list of arrays are displayed for the selected layer, and the user can select arrays to edit. When done editing arrays for that layer, the layer-number screen is again displayed. The user can choose another layer, but ultimately the user will want to quit editing hydraulic data. The F4 command is available on the layer-number screen for this purpose.

Key	Command	Command Description
F5	Limits	Creates an optional window that shows the limits for values that can be entered into the data field in which the current cursor is located. In addition to minimum and maximum values, a "Default" value is sometimes shown. The Default has no function within MFI, and it should be ignored. The Default is usually shown as “None.” Note that the Default is NOT substituted when a carriage return is entered for a field’s value. Entering a carriage return as a field's value results in the field keeping the value that is currently displayed. When a screen is activated, most data fields are given a starting value. The starting value is the previously entered value for that field, or if there is no previous value, MFI generates a starting value. A field’s starting value has no relation to the “Default” value in the limits window.
F8	Quiet	Deletes whatever optional window is being displayed (Limits, Help, or Scratch Pad), making more lines available for the data window. This command is only available when an optional window has been activated.
F9	Xpad	Creates an optional scratch pad window that allows the user to write notes. The notes are stored in the file XPAD.DAT. After entering notes, use F3 to continue editing data.
	Oops	Restarts the current screen with data values set equal to their values when the screen was last started. The only way to access this command is through the F3 command.
Page down	Dnpg	When a window contains more lines of information than can be displayed, scrolling is allowed in order to allow all lines to be seen. If there are lines below the bottom of the window, Dnpg can be used to scroll downward. The down arrow key can also scroll a line at a time.
Page up	Uppg	When a window contains more lines of information than can be displayed, scrolling is allowed in order to allow all lines to be seen. If there are lines above the top of the window, Uppg can be invoked to scroll upward. The up arrow key can also scroll a line at a time.
	Window	When a list (or table) of data is being entered using an MFI screen, the Window command is used to change more than one value at a time. To do this, move the cursor to the first value to be changed, and activate the Window command. Then use the cursor keys to highlight the block of values to be modified. After highlighting, enter the desired data value, and press the “Enter” key. The only way to access this command is through the F3 command.

Main Menu

Once the dataset is established, the Main menu is displayed as shown in figure 2. This menu contains two sets of options -- “Data Editing Options” and "Other Options." The EXIT option in this menu is the only way to exit the program normally. In order to generate a correct set of files for MODFLOW and MODPATH, it is important to exit through this option. The EXIT option causes the files to be written.

```
MAIN Menu

Data Editing Options                         Other Options
--------------------                         -------------
BAS     -- Edit Basic Package data           ACTIVE  -- Display active options
BCF     -- Edit Block-Centered Flow data     DEACT.  -- Deactivate options
STRESS  -- Edit stress data                  FILES   -- Display files being used
SOLVER  -- Edit solver data                  SaveAs  -- Save data using new name
OUTPUT  -- Edit Output Control data          EXIT    -- Save data and exit
MODPATH -- Edit MODPATH data

HELP

    Activates the Basic Package and allows data editing for this package.

INSTRUCT
                    Select an option using arrow keys
                 then confirm selection with the F2 key,  or
                   Type the first letter of an option.

Help:F1   Accept:F2   Quiet:F8   Xpad:F9
```

Figure 2. -- Main menu of MFI.

The data editing options are generally structured according to MODFLOW's primary options. The Basic Package, Block-Centered Flow Package, and Output Control Option are separate options in the Main menu. MODPATH is also an option. The stress packages are combined into a single Main Menu option, "STRESS." The three solvers for which MFI can prepare data are included in the single "SOLVER" option. Note that if the user does not select a solver, MFI will automatically select one.

There are two information options in the Main menu. The FILES option displays all files that have been defined and shows their use. The ACTIVE option shows which of MODFLOW's primary options are active. In addition, the ACTIVE option shows whether or not each active option has all of its data defined. If an option is designated as "incomplete," then some of its data have not been specified. The specific undefined data can be displayed for an option by selecting the appropriate editing option from the Main menu, and then selecting the "Undefined" option. For example, if Block-Centered Flow (BCF) data are incomplete, select BCF from the Main menu. Within the BCF Menu is an Undefined option, which will display all undefined parameters for the BCF Package. The user may be surprised that some data that have not been entered do not show as undefined. This is because MFI automatically specifies defaults for many values. These values can be changed if desired, but they need not be explicitly defined by the user if the defaults are adequate.

The “SaveAs” menu item is used to create a dataset with a new name. This is useful for making a modified dataset while keeping the original. When SaveAs is used, a copy of the current dataset is made using the new name. If the original dataset existed prior to invoking MFI, it will be left unchanged. When MFI is exited, the final version of the dataset is again saved using the new name.

Data Editing Menus

Selecting one of the data editing options in the Main Menu activates a data editing menu screen. All of the data editing menu screens have a DONE option, which will return to the Main menu. As noted earlier, the user must ultimately end up back in the Main menu in order to exit MFI. The data editing menus contain appropriate options for entering the required data. The data entry screens can be nested one or more levels depending on the complexity of the data. The selection of a data editing option automatically activates that option in MODFLOW or MODPATH.

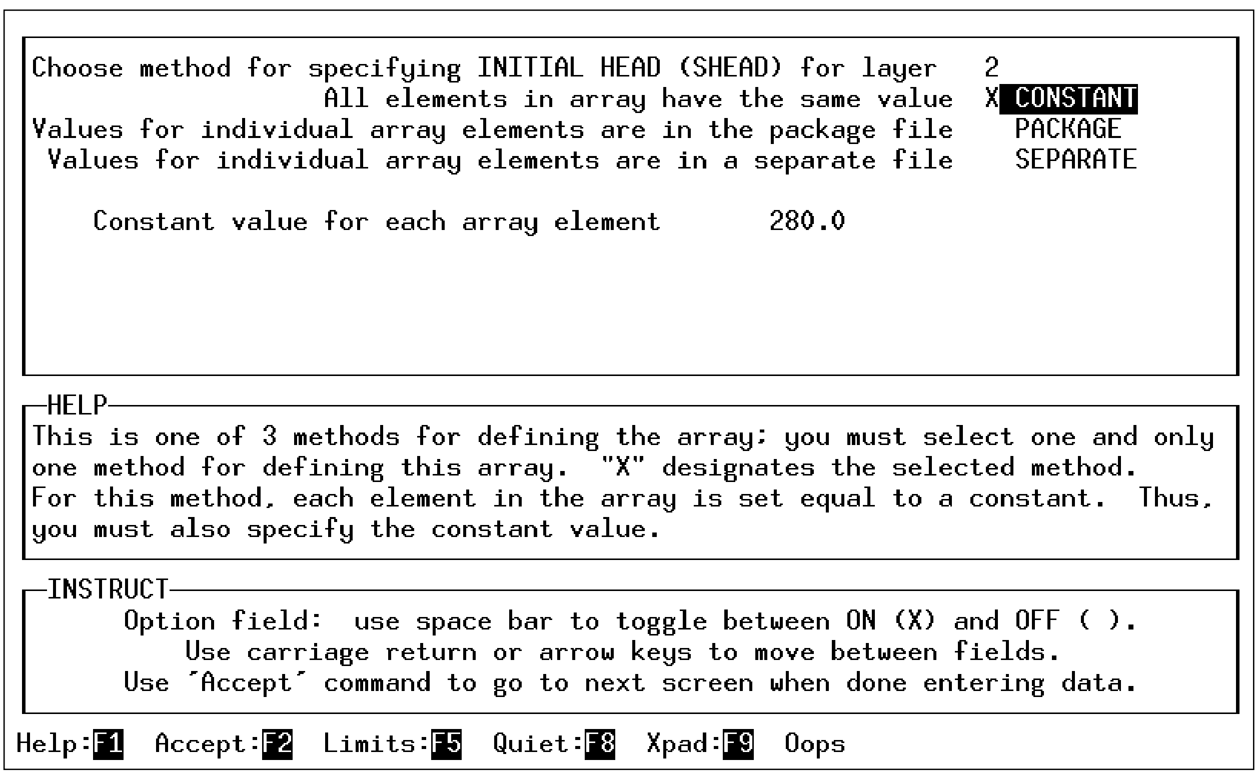
The DEACTIVATE option is provided in the Main Menu in order to deactivate any undesired options that have been previously activated. If an option is deactivated, any changes that had been made to its data during the current MFI session will not be saved when MFI exits, and MODFLOW and MODPATH will not include that option when they are subsequently executed. However, if a file for a deactivated option existed prior to its deactivation, that file will still exist after MFI exits.

When entering data for the entire grid, such as hydraulic parameters, the data are entered a layer at a time. Before entering such data, the user will be prompted for the layer number unless the model consists of a single layer. The only way to stop entering data for layers is to use the Prev (F4) command when asked for a layer number, which returns to a menu.

When entering stress data for simulations with more than one stress period, the user will be prompted for the stress period prior to prompting for the actual stress data. When asked for a stress period, the only way to stop entering data for stress periods is to use the PREV (F4) command, which returns to a menu.

Entering Two-Dimensional Arrays

There are three methods that can be used to define two-dimensional arrays. These correspond to the three methods that MODFLOW and MODPATH use for defining arrays. The first method, referred to as the constant method, is to make the entire array a constant. The second, or package, method is to store individual array elements in the appropriate MODFLOW package file (or MODPATH file) along with other data. When the package method is used, MFI allows the values for individual array elements to be entered or modified by invoking a spreadsheet program. MFI writes the array-element values in the appropriate package file when exited. For an existing dataset, the initial values for a package array are read from the appropriate package file. The third, or separate, method is to specify that the values for individual array elements are in a separate file, meaning a file other than a package file. MFI does not actually read or write the contents of such files, but MFI will incorporate information about these files into the proper locations in the dataset so that MODFLOW and MODPATH will read them. The separate files must be prepared using other software.



```
Choose method for specifying INITIAL HEAD (SHEAD) for layer   2
                   All elements in array have the same value  X CONSTANT
Values for individual array elements are in the package file    PACKAGE
 Values for individual array elements are in a separate file    SEPARATE

    Constant value for each array element       280.0

HELP
This is one of 3 methods for defining the array; you must select one and only
one method for defining this array.  "X" designates the selected method.
For this method, each element in the array is set equal to a constant.  Thus,
you must also specify the constant value.

INSTRUCT
      Option field:  use space bar to toggle between ON (X) and OFF ( ).
          Use carriage return or arrow keys to move between fields.
      Use 'Accept' command to go to next screen when done entering data.

Help:F1  Accept:F2  Limits:F5  Quiet:F8  Xpad:F9  Oops
```

Figure 3. -- Array screen defining a CONSTANT array.

All three methods of defining arrays are specified through one screen. Figure 3 shows an example array screen in which a constant array is being defined. In that example, the "CONSTANT" option is indicated by an "X." The other two options, "PACKAGE" and "SEPARATE," can be selected by moving the cursor to the desired option and pressing the space bar, as indicated by the Instruction Window. Figure 4 shows the PACKAGE option, and Figure 5 shows the SEPARATE option. Notice that for each option the other data fields change. For PACKAGE and SEPARATE arrays, a constant is specified that is multiplied times each element of the array to make the final value. Thus, the entire array can be rescaled simply by changing the multiplier.

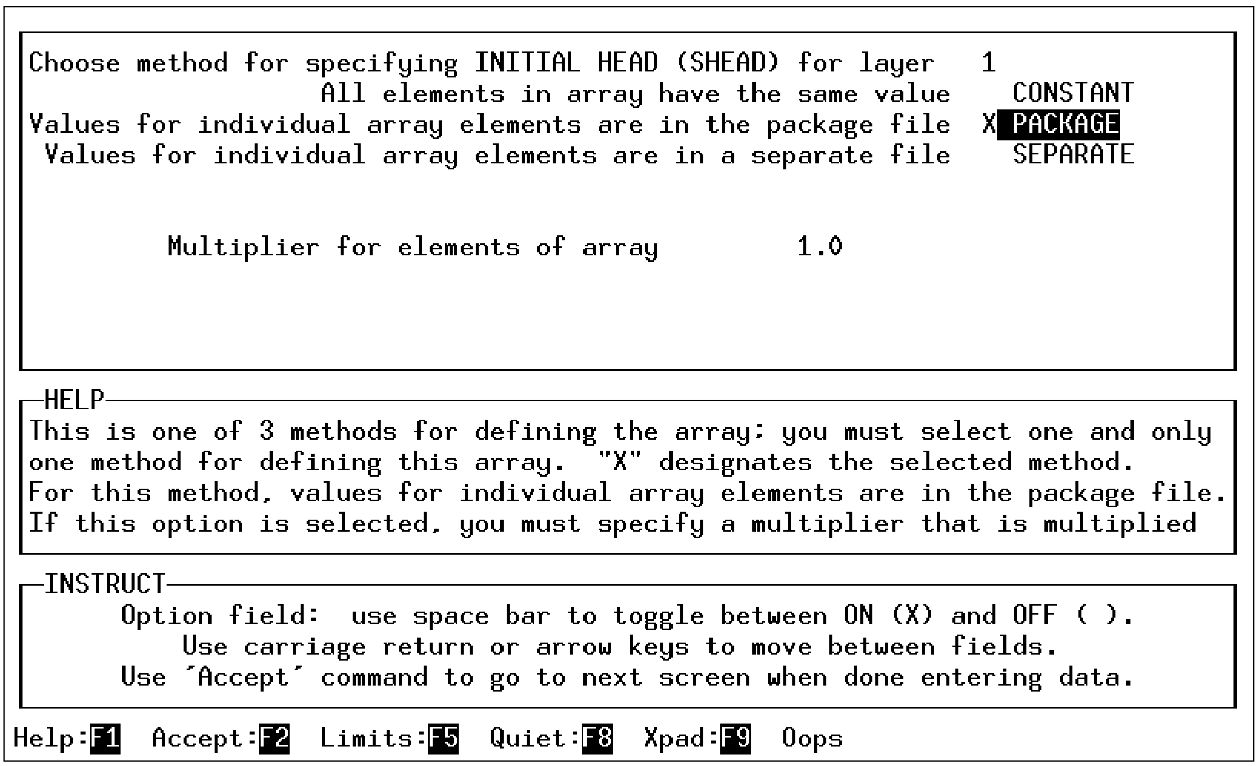


Figure 4. -- Array screen defining a PACKAGE array.

```
Choose method for specifying BOT    for layer   1
                   All elements in array have the same value    CONSTANT
Values for individual array elements are in the package file    PACKAGE
 Values for individual array elements are in a separate file  X SEPARATE


         Multiplier for elements of array        1.0
File Name: bottom.dat
FORMAT of data in separate file (FMTIN)  (10F8.0)
Code for printing the data after MODFLOW reads it (IPRN)      0

HELP
This is one of 3 methods for defining the array; you must select one and only
one method for defining this array.  "X" designates the selected method.
For this method, values for individual array elements are in a separate file.
If this option is selected, you must specify a multiplier that is multiplied

INSTRUCT
      Option field:  use space bar to toggle between ON (X) and OFF ( ).
          Use carriage return or arrow keys to move between fields.
      Use 'Accept' command to go to next screen when done entering data.

Help:F1  Accept:F2  Limits:F5  Quiet:F8  Xpad:F9  Oops
```

Figure 5. -- Array screen defining a SEPARATE array.

After an array is first designated as a PACKAGE array, MFI activates the spreadsheet program to allow the array elements to be entered. Once a PACKAGE array has been initially defined, its elements can be examined or modified using the spreadsheet program through the same array screen. However, prior to activating the spreadsheet program to edit an existing PACKAGE array, the user is asked if he or she wants to edit the array or leave the values of each element as they are. If the values are to be edited, then the spreadsheet is activated. The choice to leave the array unedited is offered because there may be no need to change these values. That is, the reason for editing the array may have been only for the purpose of changing the array multiplier.

For SEPARATE arrays, the information that will allow MODFLOW or MODPATH to read the array must be specified. This includes the file name, a format, and a code for which format to use when printing the array after it is read. Remember that MFI does not read SEPARATE arrays, so if there are errors, such as an incorrect format or an invalid file name, they will not be detected by MFI. Such errors will not be detected until MODFLOW or MODPATH is executed.

When a MODPATH array is in a separate file and if that file is first used interactively (rather than as part of an existing dataset), MFI asks if the file will contain only a single array or if the file will contain multiple arrays. If it contains only a single array, then it will be setup as an OPEN/CLOSE file (Pollock, 1994, Appendix A). If the file contains multiple arrays, then it will be setup as an EXTERNAL file (Pollock, 1994, Appendix A). MFI maintains information about each file used, which includes a file type. OPEN/CLOSE files are file type OPEN/CLOSE, and EXTERNAL files are file type DATA. The screen asking whether there will be a single array or multiple arrays in a file only appears when a file is added to the list of files, thus all previous uses of a file must be temporarily ended so the file gets removed from the list of files if there is a need to change a file's type. Then the file can be recreated with a new file type. For example, consider a separate file named por.dat that initially is expected to contain only a single porosity array for MODPATH. It might therefore be given file type OPEN/CLOSE. If the user later decides to incorporate two porosity arrays in por.dat, the file type needs to be changed to DATA. To do this, the prior use of the file must be eliminated. The easiest way to do this is to change the array that was previously read from por.dat to be a constant. Unused files are removed from the list of files only when the Main Menu is entered, so the next step must be to exit the MODPATH Menu. Then the MODPATH Menu can be reentered, and por.dat can be specified as a separate file that will contain multiple arrays.

Entering Data with the Spreadsheet Program

MFI uses a separate spreadsheet program to allow the user to enter two types of data. One type is data that corresponds to every cell throughout a model layer such as transmissivity. The values are displayed and edited as a spreadsheet table; the cell locations are implicitly designated by the row and column locations in the table. The second type of data for which a spreadsheet is used is a list of data in which each entry in the list includes a layer, row, and column grid location plus one or more parameters that apply at that location. An example of such a list is the list of wells for a particular stress period. For each well, there are four values, which are the three grid coordinates plus the well recharge rate (negative value indicates pumping). If MODPATH is being used, each entry also includes a code to indicate the face through which the stress is applied. The spreadsheet includes column headings that define the parameters to be entered.

The spreadsheet program is automatically activated when needed. Once activated, the user edits the data, saves it, and quits the spreadsheet using the spreadsheet commands. When the spreadsheet program quits, MFI reads the data saved by the spreadsheet and continues processing.

As documented in this report, MFI uses the Tactician Spreadsheet program by Soft-tek. User instructions for Tactician are not included here; it is assumed that users of MFI will refer to appropriate user manuals for this information. The file "tact.t" is temporarily used to hold the data being edited by the spreadsheet. When done editing the data, the user must save the modified data into the same "tact.t" file. Then the user should exit the spreadsheet. After the spreadsheet program quits, MFI reads the modified data from "tact.t" and deletes that file.

MFI would have to be modified in order to use a different spreadsheet program. Information about program modification is contained in the Program Documentation Section.

Obtaining Help When Using MFI

To make it easy to identify the specific parameters being entered, the MFI data screens describe each parameter. In addition, the help command (F1) causes a help window to be displayed. The help window provides information about the individual menu choices or data parameters of a screen. If the help information is inadequate, then the MODFLOW or MODPATH documentation should be consulted. The screen descriptions usually indicate the actual parameter names used in the MODFLOW and MODPATH input instructions. For MODFLOW, the data editing menus generally correspond to primary options in the MODFLOW documentation (McDonald and Harbaugh, 1988). Details about the data parameters can be found in the input instructions in the corresponding chapter of the documentation. For MODPATH, details about input parameters can be found in Appendix A of Pollock (1994).

Data Recovery if MFI Aborts

If something goes wrong and MFI aborts, some data may be recoverable. If the error is detected by MFI, it attempts to save all current data in files named "MFI" followed by a period and a two- or three-character option designation, for example, "MFI.RCH" for the recharge option. Even if MFI aborts without detecting a cause -- for example, the power to the computer could be shut off -- there may be some recoverable data. This is because every time the menu for editing data for a primary option is completed through the "DONE" option, a backup file of the data for that option is written to an “MFI...” file. However, the user would have to use a text editing program and have knowledge of the structure of the files (from MODFLOW or MODPATH documentation) in order to extract data from them.

Running MODFLOW and MODPATH

MODFLOW uses preconnected files, which means that the method of specifying the file names that correspond to Fortran's numeric file units is determined by the computer vendor. On the Data General workstations on which MFI was developed, it is required that preconnected files be named "fort.n" where n is the unit number. Thus, prior to running MODFLOW, the "fort.n" files must be created from the actual data files that are specified within MFI. MFI does this by writing a Unix shell program that creates the files, executes MODFLOW, and then deletes the "fort.n" files. The shell program is the dataset name followed by ".sh". Thus, once all data are entered and MFI is exited, MODFLOW can be run by running the shell program. Modifications to MFI may be necessary in order to meet the requirements for preconnected files on other computers. The Program Documentation Section gives information about where the modifications would be made.

MODPATH obtains information about the files that it will use from the MODPATH name file. MFI creates the MODPATH name file, and its name is the dataset name followed by ".mpn".

Although MFI creates datasets of the proper form for MODFLOW and MODPATH, it is possible for incorrect data to be entered using MFI. MFI does a lot of data checking in order to reduce the chance for errors, but the checking is not exhaustive. Further, no amount of checking can detect data entry errors in which a reasonable, but incorrect, value is mistakenly entered. Some errors will cause MODFLOW or MODPATH to stop before completion while others will result in what might first appear to be normal execution, but the results are wrong. Thus, the output files from MODFLOW and MODPATH should always be examined to make sure that data have been defined as intended.

Miscellaneous Details About Using MFI

Parameters that MFI Calculates

Those already familiar with MODFLOW or those who carefully examine the MODFLOW and MODPATH input instructions will notice that the MFI data screens do not directly include all parameters required for MODFLOW and MODPATH. This is because MFI calculates some parameters either by assumption or from other requested information. Table 2 lists the parameters that are calculated by some method other than direct data entry.

Table 2. -- Parameters that MFI calculates

Option	Parameter	How Calculated
BAS	IAPART	0 if LAYAVG (Goode and Appel, 1992) is not 30; 1 if LAYAVG is 30.
BAS	ISTRT	0 when drawdown is not calculated; 1 when drawdown is calculated.
BAS	IUNIT	Unit numbers for files are automatically selected.
BAS	NPER	User specified when transient; 1 when steady state.
BAS	PERLEN	User specified when transient; 1.0 when steady state.
BAS	NSTP	User specified when transient; 1 when steady state.
BAS	TSMULT	User specified when transient; 1.0 when steady state.
BAS	ITMUNI	User specified when transient; set to “UNDEFINED” when steady state.
BCF	ISS	Determined from the “Steady State/Transient” option specified in the Miscellaneous input screen for the Basic Package; 0 if transient, and 1 if steady state.
BCF	IBCFCB	Determined from data entered in Output Control screens.
RIV	ITMP	Determined by counting the reaches entered by the user.
RIV	MXRIVR	Determined by looking at the number of reaches used in each stress period and finding the maximum number.
RIV	IRIVCB	Determined from data entered in Output Control screens.
WEL	ITMP	Determined by counting the wells entered by the user.
WEL	MXWELL	Determined by looking at the number of wells used in each stress period and finding the maximum number.
WEL	IWELCB	Determined from data entered in Output Control screens.
DRN	ITMP	Determined by counting the drains entered by the user.
DRN	MXDRN	Determined by looking at the number of drains used in each stress period and finding the maximum number.
DRN	IDRNCB	Determined from data entered in Output Control screens.
GHB	ITMP	Determined by counting the boundaries entered by the user.
GHB	MXBND	Determined by looking at the number of boundaries used in each stress period and finding the maximum number.
GHB	IGHBCB	Determined from data entered in Output Control screens.
RCH	INRECH	-1 if RECH array is reused from previous stress period; otherwise 0.
RCH	INIRCH	-1 if IRCH array is reused from previous stress period; otherwise 0.
RCH	IRCHCB	Determined from data entered in Output Control screens.

Option	Parameter	How Calculated
EVT	INIEVT	-1 if IEVT array is reused from previous stress period; otherwise 0.
EVT	INEVTR	-1 if EVTR array is reused from previous stress period; otherwise 0.
EVT	INEXDP	-1 if EXDP array is reused from previous stress period; otherwise 0.
EVT	INSURF	-1 if SURF array is reused from previous stress period; otherwise 0.
EVT	IEVTCB	Determined from data entered in Output Control screens.
SIP	NPARM	Always 5.
SIP	IPCALC	0 if WSEED is not 0; 1 if WSEED is 0
PCG	NPCOND	Always 1
PCG	NBPOL	Always 2.
PCG	IPCGCD	Always 0.
Output	IHEDUN	0 if head is not saved. If head is saved, unit number is automatically selected.
Output	IDDNUN	0 if drawdown is not saved. If drawdown is saved, unit number is automatically selected.
Output	INCODE	Determined from data entered in Output Control screens.
Output	IHDDFL	Determined from data entered in Output Control screens.
Output	IBUDFL	Determined from data entered in Output Control screens.
Output	ICBCFL	Determined from data entered in Output Control screens.
Output	Hdpr	Determined from data entered in Output Control screens.
Output	Ddpr	Determined from data entered in Output Control screens.
Output	Hdsv	Determined from data entered in Output Control screens.
Output	Ddsv	Determined from data entered in Output Control screens.
Modpath	IGRID	Always 0.
Modpath	NCBL	Calculated by counting the number of layers for which NCON (the confining bed code) is not 0.
Modpath	TOP	For layers below the top layer, TOP is defined as the BOT of the layer above when there is no confining layer above; otherwise TOP is either user defined or the elevation of the water table.

Parameters Shared Between MODFLOW and MODPATH

Many parameters are shared between MODFLOW and MODPATH. In MFI, these shared parameters are entered only within MODFLOW's data editing screens. That is, the shared parameters are not included as part of the screens used for MODPATH's data. These shared parameters are:

NLAY
NROW
NCOL
NPER
NSTP
PERLEN
TSMULT
HNOFLO
HDRY
LAYCON
DELR
DELC
TOP for layer types 2 and 3
BOT for layer types 1 and 3
Names of budget file, head file, and data files for stress packages

Note that IBOUND is not shared by MODPATH and MODFLOW even though both MODFLOW and MODPATH have a parameter named IBOUND. The reason is that MODPATH makes expanded use of IBOUND. In addition to indicating boundary conditions as in MODFLOW, MODPATH uses IBOUND to indicate zones for the purpose of controlling which particles and pathlines are displayed and their colors. There are situations in which different IBOUND values may be desired for MODFLOW and MODPATH. Thus, these are viewed as separate arrays, except that when MODPATH is first activated, any MODFLOW IBOUND arrays that are constant or package arrays are copied to MODPATH's IBOUND. Further, whenever IBOUND is modified in MODFLOW or MODPATH, the values are checked for consistency (no flow, constant head, and variable head at the same locations). If there are inconsistencies, the user is notified, but nothing is done to attempt to remedy the situation. No consistency checking is done for any layers for which IBOUND is specified as being in a separate file in either MODFLOW or MODPATH.

MODPATH includes the ability to specify which cell face receives a particular stress. This capability is incorporated in MFI through the screens in which stress data are entered. When MODPATH is active, the stress screens include an extra parameter to indicate the cell face for the stress. For the River, Well, Drain, and General-head Boundary stresses, the stress face is specified in the list of stresses each stress period. For Recharge and Evapotranspiration stresses, the face is specified as a single value for all cells for the entire simulation. Thus, it is more efficient to activate MODPATH prior to entering stress data if the user knows that MODPATH will eventually be used.

This will allow the stress faces to be entered when the other stress data are entered. If stress data are entered prior to activating MODPATH, then the stress screens will need to be reentered after MODPATH is activated if there is a need to specify specific faces to recieve stresses.

Constraints in MFI that are not in MODFLOW and MODPATH

Ideally, MFI should impose no constraints on data that MODFLOW and MODPATH do not impose; however, some compromises were made in this regard. The additional constraints imposed by MFI are listed below. There are also some configurable limits, which relate to the amount of memory used by MFI, that are not included in this list of constraints. See Program Documentation Section for information about the configurable limits.

1. MFI does not allow a steady-state simulation to have more than one stress period.

2. The precision of data values in files created by MFI is controlled by MFI. In most cases, values will be written with at least four significant digits. If the precision used is inadequate, MFI cannot be used or must be modified.

3. If output does not occur at regular intervals (every time step, end of every stress period, or end of simulation), then a maximum of 47 times can be entered at which output will occur.

4. If specific layers for printing or saving head or drawdown are listed rather than using the default of having this information output for all layers, a maximum of 15 layers can be specified.

5. When MODPATH is used, each TOP and BOT array that is in a separate (external) file must be in a file by itself. This constraint is needed in order to insure that MODFLOW and MODPATH can share the same arrays. MODFLOW and MODPATH cannot share multiple TOP and BOT arrays from the same external file because they read these arrays in a different order.

6. A maximum of 92 file units can be used for MODFLOW files -- units 1-92. A maximum of 77 file units can be used for MODPATH files -- units 1-4 and 7-79.

Using MFI with an Existing Dataset

MFI can work with MODFLOW files that were not originally created using MFI. To use existing MODFLOW data, the user must create an MFI name file. The name file must end with ".nam" and have a one to eight character prefix, which will be the MFI dataset name. The name file should include all files used by MODFLOW. Each line in the name file includes a file type, a file unit, and a file name. The file type is "LIST" for the listing file, "DATA" for a separate formatted file, "DATA(BINARY)" for a separate binary (unformatted) file, or the name of a primary option such as RCH. The order of entries does not matter. The “DATA(BINARY)” file type is used primarily for output files (head, drawdown, and budget). Units from 1 through 300 are allowed. The entries in the name file are read using free format, so spacing is not important as long as a comma or at least one space separates adjacent fields. An example name file is:

```
LIST    6   test.lst
BAS     1   test.bas
RCH    18  test.rch
OC    22  test.oc
PCG    23  test.pcg
DATA(BINARY)  50  test.hed
DATA  60  transmissivity.1
```

The MODFLOW input data also must adhere to two rules. First, for most parameters, a value of zero must be specifically represented with a zero character rather than using a field of all blanks. MFI will generally not interpret a blank field as a value, and the next non-blank field will be used instead. After this happens, the remaining data will be read incorrectly. Second, adjacent fields must be separated by one ore more spaces. Two fields that are not separated will typically be interpreted by MFI as a single value. As data are read, they are written to the log file; so the user can examine the log file in order to find where these errors have occurred.

MFI can also read MODPATH data that were created without MFI. In this situation, a MODPATH name file would already exist. This name file must be named with the MFI dataset name followed by “.mpn.”

Using MFI When MODFLOW Includes Packages that MFI Does Not Include

MFI is not able to prepare data for all packages that have been created for MODFLOW. If one is using such packages, MFI can still be used for those packages that it does include. The data for the other packages will of course have to be prepared by other methods. Also, the Basic Package file must include the unit number for all of the other package files (McDonald and Harbaugh, 1988, p. 4-11). The user can manually make this change to the Basic Package file after MFI is run each time, or MFI can be modified to recognize package names that it does not incorporate and include their units in the Basic Package file. See the Program Documentation Section for a list of packages that are recognized by MFI and for information about modifying MFI to recognize other packages.

PROGAM DOCUMENTATION

The source code for MFI is not included in this report; it can be obtained as described in the report preface. The information in this section is for the purpose of helping someone who has obtained the source code to be able to adapt the program to run on different computers.

Memory Usage

MFI stores the data that it reads in memory. Like MODFLOW, MFI uses a single array, X, to store most of its data. Locations of various parameters within X are maintained by MFI. Instead of having a separate variable that locates each parameter in X, MFI maintains a list of parameter names and their locations. The list is searched whenever the location of a parameter is needed.

Lists of stress data for a stress period require special consideration because MFI does not require the maximum size of a list to be specified. MFI allows the user to enter list data of varying length, and the length can change at any time. In order to store lists of varying size, MFI allocates memory for lists in blocks. Each block has a fixed size, but a list can consist of up to 10 blocks.

The following parameters control how much memory MFI can use. These parameters are defined in the file “mfi.gbl,” which is an include file used in many of MFI’s subroutines. In addition to the size of the X array, these parameters specify the sizes of a number of other arrays used by MFI to keep track of data.

LENX -- Number of elements in the X array
MXVAR -- Maximum number of parameters
MXARR -- Maximum number of parameters that are arrays
MXFNAM -- Maximum number of files
MXLIST -- The number of entries in a list block; a list can consist of up to 10 blocks

Computer Dependencies

Although MFI is designed for portability among computers, there are a number of computer dependencies that cannot be avoided. These dependencies are isolated so that the code that must be changed in order to adapt MFI for use on another computer can be easily found.

The largest section of code that is computer dependent is the AIDE library. As explained in the Preface, this code is available for three computer systems: Data General Unix workstations, Sun Sparcstations, and IBM compatible personal computers. It may be possible for a knowledgeable programmer to convert this code for use on other computers.

The code for interfacing to the spreadsheet program is contained in file “spread.f”. Three subroutines are required. Subroutine SSLST uses the spreadsheet program to edit a list of data. Subroutine SS2DI edits a 2-dimensional integer array, and Subroutine SS2DR edits a 2-dimensional real array. Two versions of spread.f are supplied with the MFI source code. One version is for the Tactician spreadsheet program, and the other is for the Lotus 123 spreadsheet program. It should be possible to adapt these subroutines for other spreadsheet programs; however, this requires a skilled programmer who can figure out how data are stored within MFI and in files that the spreadsheet can read and write.

The file “system.f” contains system dependent subroutines. Subroutine SYSCAL calls the operating system to execute a command. This is how the spreadsheet is activated. Subroutine NAMARG gets a command line argument, which is used for the purpose of getting the dataset name from the command that invokes MFI. The version of this subroutine included with the source code is for use on a Data General workstation. Subroutine GETPNM looks for a file name that may not be in the local directory. This is used to find the AIDE message file that is used to generate screens. This routine looks for a file named “MFI.PTH” in the current directory. If found, the first line of this file is used as the directory in which the message file can be found. If MFI.PTH is not found, then it is assumed that the message file is in the current directory. Some computers provide special routines that scan a set of directories in order to locate a file; GETPNM could be modified to take advantage of such a routine. Subroutine ORGOPN writes a file of commands necessary to connect the files used by MODFLOW to Fortran’s numeric file units. As described in the Running MODFLOW and MODPATH Section, this routine creates a shell program for Data General workstations.

Modifying MFI to Specify Which Options Use Which File Units

In MODFLOW, the IUNIT array specifies which options are active and the file units used to read the input data for those options. Many of the IUNIT members were assigned to specific options in McDonald and Harbaugh (1988), but other options have been documented with conflicting IUNIT assignments. Thus some versions of MODFLOW may use different IUNIT values for some options. The MFI file units must match the values used in MODFLOW. Also, the units used for the Basic Package and the listing file are specified in the main MODFLOW program. These may vary according to the specific computer used. The following list shows the file unit assignments for all the options of MODFLOW that are included in MFI as well as for the Basic Package and listing file.

unit 5 -- Basic Package file
unit 6 -- Listing file
IUNIT(1) -- Block-Centered Flow Package, version 2 (McDonald and others, 1992) and version 3 (Goode and Appel, 1992)
IUNIT(2) -- Well Package
IUNIT(3) -- Drain Package
IUNIT(4) -- River Package
IUNIT(5) -- Evapotranspiration Package
IUNIT(7) -- General-Head Boundary Package
IUNIT(8) -- Recharge Package
IUNIT(9) -- Strongly Implicit Procedure Package
IUNIT(11) -- Slice-Successive Overrelaxation Package
IUNIT(12) -- Output Control Option
IUNIT(13) -- Preconditioned Conjugant Gradient Package (Hill, 1990)

Further, MFI recognizes several options for which it cannot prepare input data. MFI will put the proper unit number into the IUNIT array (in the Basic Package file) for these options even though it will not actually prepare the data files. The IUNIT assignments for these additional options are:

IUNIT(6) -- Transient Leakage (Leake and others, 1994)
IUNIT(14) -- Generalized Finite-Difference Package (Harbaugh, 1992)
IUNIT(16) -- Horizontal Flow Barrier Package (Hsieh and Freckleton, 1993)
IUNIT(18) -- Stream Package (Prudic, 1989)
IUNIT(19) -- Interbed Storage Package (Leake and Prudic, 1991)
IUNIT(20) -- Time-Varying Specified Head Package (Leake and Prudic, 1991)

To change the unit assignments, the CUNIT array in the MFI main program must be modified. The CUNIT array contains a 3-character file type that corresponds to each IUNIT element. For example, IUNIT(2) corresponds to the Well Package, and CUNIT(2) has the value “WEL.” The CUNIT values that correspond to unused IUNIT elements should be blank. Note that MODPATH does not have an IUNIT assignment within MODFLOW because MODPATH is a separate program; however, within MFI, MODPATH is assigned to IUNIT(40) with a file type of “MP.”

REFERENCES CITED

Goode, D. J. and Appel, C. A., 1992, Finite-difference interblock transmissivity for unconfined aquifers and for aquifers having smoothly varying transmissivity: U.S. Geological Survey Water-Resources Investigations Report 92-4124, 79 p.

Harbaugh, A. W., 1992, A generalized finite-difference formulation for the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 91-494, 60 p.

Hill, M. C., 1990, Preconditioned conjugate-gradient 2 (PCG2), a computer program for solving ground-water flow equations: U.S. Geological Survey Water-Resources Investigations Report 90-4048, 43 p.

Hsieh, P. A. and Freckleton, J. R., 1993, Documentation of a computer program to simulate horizontal-flow barriers using the U.S. Geological Survey's modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 92-477, 32 p.

Kittle, J. L., Jr., Hummel, P. R., and Imhoff, J. C., 1989, ANNIE-IDE, A system for developing interactive user interfaces for environmental models (Programmers Guide): U.S. Environmental Protection Agency, EPA/600/3-89/034, 166 p.

Leake, S. A., Leahy, P. P., and Navoy, A. S., 1994, Documentation of a computer program to simulate transient leakage from confining units using the modular finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 94-59, 70 p.

Leake, S. A. and Prudic, D. E., 1991, Documentation of a computer program to simulate aquifer-system compaction using the modular finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water Resources Investigations, Book 6, Chapter A2, 68 p.

McDonald, M. G. and Harbaugh, A. W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water Resources Investigations, Book 6, Chapter A1, 586 p.

McDonald, M. G., Harbaugh, A. W., Orr, B. R., and Ackerman, D. J., 1992, A method of converting no-flow cells to variable-head cells for the U.S. Geological Survey modular finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 91-536, 99 p.

Pollock, D. W., 1994, User's guide for MODPATH/MODPATH-PLOT, Version 3: A particle tracking post-processing package for MODFLOW, the U.S. Geological Survey finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 94-464, 6 ch.

Prudic, D. E., 1989, Documentation of a computer program to simulate stream-aquifer relations using a modular, finite-difference, ground-water flow model: U.S. Geological Survey Open-File Report 88-729, 113 p.