User's Documentation for MODFLOW-96, an update to the U.S. Geological Survey Modular Finite-Difference Ground-Water Flow Model

by Arlen W. Harbaugh and Michael G. McDonald

U.S. GEOLOGICAL SURVEY

Open-File Report 96-485

Reston, Virginia
1996

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 20192

Copies of this report can be
purchased from:

U.S. Geological Survey
Branch of Information Services
Box 25286
Denver, CO 80225-0286
(303) 202-4700

PREFACE

This report presents a revised version of the U.S. Geological Survey (USGS) modular finite-difference ground-water flow model, commonly known as MODFLOW. The program has been tested by using it for a variety of model simulations, but it is possible that other applications could reveal errors. Users are requested to notify the USGS if errors are found in this report or the program.

Although this program has been used by the USGS, no warranty, expressed or implied, is made by the USGS or the United States Government as to the accuracy and functioning of the program and related program material. Nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

The computer program is available through the World-Wide Web at address

http://h2o.usgs.gov/software/

or by anonymous ftp file transfer from directory /pub/software/ground_water/modflow at Internet address

h2o.usgs.gov

The computer program is also available on diskette for the cost of processing from:

U.S. Geological Survey
NWIS Program Office
437 National Center
Reston, VA 20192
Telephone: (703) 648-5695

CONTENTS

Page

Abstract 1

Introduction 2

Description of changes 4

Highly visible changes 4

Opening files 4

Specifying data read by the array-reading modules 4

Specifying data that are not read by the array-reading modules 5

Listing file 5

Greater flexibility in generating files for interfacing with other programs 6

Output control 6

Cross-sectional models 7

Less visible changes 7

Precision of budget calculations 7

Ability to add simulation capabilities 8

Miscellaneous changes 9

Compatibility with code added by others 10

Input Instructions 12

Name File 13

Basic Package 15

Output Control Option 18

Block-Centered Flow Package 22

River Package 26

Recharge Package 28

Well Package 29

Drain Package 31

Evapotranspiration Package 33

General-Head Boundary Package 35

Strongly Implicit Procedure Package 37

Slice-Successive Overrelaxation Package 38

Utility Modules 39

Example simulation 42

References 56

TABLES

Page

Table

1-8. Example input data for:

1. Name file 42
2. Basic Package 43
3. Block-Centered-Flow Package 44
4. Well Package 44
5. Drain Package 44
6. Recharge Package 45
7. Strongly Implicit Procedure Package 45
8. Output Control Option 45

9. Output listing from example problem 46

USER'S DOCUMENTATION FOR MODFLOW-96, AN UPDATE TO THE U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER FLOW MODEL

By Arlen W. Harbaugh[1] and Michael G. McDonald[2]

ABSTRACT

A number of changes have been made to the U.S. Geological Survey modular finite-difference ground-water flow model, which is commonly known as MODFLOW. Existing MODFLOW input files will work with the revised model. Also, the basic structure and computational methods of the model have been maintained. Those familiar with the original model should have no difficulty adapting to the revised model. Some of the most significant changes are:

1. All files are opened with Fortran OPEN statements. The user must prepare an additional file that specifies the model files to be opened.

2. The array-reading subroutines have been enhanced to make it easier to prepare the array control records. When using the new control record, all the parameters are read using free format; however, the original fixed-format control records can still be used if desired.

3. For input not read with array-reading subroutines, an option to use free format has been added. This avoids the need to place data values in specified columns of input records, and thus will prevent the common error of entering data in incorrect columns. If this option is not used, data are read using the original fixed format fields.

4. The capability to produce an output listing that is 80 columns wide (rather than 132) has been added in order to facilitate printing on 8.5-inch wide paper and viewing on a video display screen. All the miscellaneous output has been reduced to 80 columns, and several new formats that are 80 columns wide have been added to the array-printing subroutines.

5. A new method for specifying output control has been added. This new method uses words rather than numeric codes for specifying the various output options. New options have also been added such as the ability to save calculated heads in formatted files and save budget data in more compact unformatted files. The old method of specifying output control can still be used.

6. Budget calculations have been changed to double precision, which improves the accuracy of budget calculations. Inaccuracy of budget calculations was sometimes noticeable in the original model when a grid of 100,000 or more cells was used or when a large conductance was used in a head-dependent stress term.

[1] U.S. Geological Survey, Reston, VA
[2] McDonald Morrissey Associates, Reston, VA

INTRODUCTION

The U.S. Geological Survey modular finite-difference ground-water flow model, which is commonly called MODFLOW, can simulate ground-water flow in a three-dimensional heterogeneous and anisotropic medium provided that the principal axes of hydraulic conductivity are aligned with the coordinate directions. A constant-density fluid is also assumed. MODFLOW was originally documented in McDonald and Harbaugh (1983) and was written in the Fortran 66 programming language (American National Standards Institute, 1966). This MODFLOW version will be referred to as MODFLOW-83. MODFLOW-83 was updated to conform to the Fortran 77 language (American National Standards Institute, 1978) in McDonald and Harbaugh (1988), and this version will be referred to as MODFLOW-88. MODFLOW-88 is functionally equivalent to MODFLOW-83 in all respects.

Over the past several years, the authors of MODFLOW have collected comments from model users about changes that would make MODFLOW more useful. A number of changes have been implemented to address the perceived needs. The revised MODFLOW will be referred to as MODFLOW-96. The changes do not involve the methods of simulating ground-water flow. MODFLOW-96 incorporates the same simulation functionality as MODFLOW-88 as well as the ability to convert dry cells to variable-head cells (McDonald and others, 1992) and the ability to calculate alternate interblock transmissivities (Goode and Appel, 1992).

The purpose of this report is to document MODFLOW-96 for readers who already have an understanding of MODFLOW-88. The documentation consists of a description of the changes to MODFLOW-88, input instructions, and a sample problem. A separate report also documents the program code (Harbaugh and McDonald, 1996). The hydrologic concepts of how the program works and the overall concepts of the modular structure are unchanged and are not repeated here.

When designing the changes, the authors followed three design goals. First, the overall structure should stay the same. This goal was met. The reason for this goal is that the modular structure is the mechanism for achieving two of the primary goals for MODFLOW: ease of understanding and ease of adding new capabilities. It would be counterproductive if MODFLOW-96 were harder to understand than earlier versions. Likewise it would be disruptive if the changes made it more difficult to make additions or made it necessary to make major revisions to the many additions that have already been made. The second goal is that input files from MODFLOW-88 should work with MODFLOW-96. This goal also was met.

The third goal is somewhat subjective but important nevertheless. This goal is that only those capabilities that cannot be acceptably implemented through a separate program should be added to MODFLOW-96. One reason for this goal is that understandability is lowered the more complex a program becomes. Another reason is that a program becomes much more difficult to create and maintain the larger it gets. Finally, greater flexibility can be obtained by creating new capabilities as separate programs that the user can combine as desired. Many capabilities depend on user preference; that is, different users often want different versions of the same functionality. There is not much chance that a single model program can be created that will contain all users' preferred ways of doing all tasks related to modeling. If "building blocks" are constructed, then users have a better chance of meeting their needs by combining the blocks as desired.

Many suggestions for improving MODFLOW include the addition of interactive data input. MODFLOW was designed as a batch (noninteractive) program, which means that it runs without user intervention by reading all input from previously prepared files. The authors believe that a computationally intensive program such as MODFLOW is best run without user interaction. A model may require hours to execute, and it is not practical for the user to sit at a computer display for long periods waiting to enter the data when needed. Of course, an interactive MODFLOW could prompt for all input data at the beginning and then start the simulation, but for the reasons stated above this same functionality could be best obtained by making a separate interactive data entry program that would prepare MODFLOW's files. Thus no interactive functionality (other than an initial prompt for a file name) was added to MODFLOW-96.

Subjectivity arises in regard to the third goal because of the use of "acceptably" in the definition of this goal. Clearly some of the new functionality implemented in MODFLOW-96 could be implemented through separate programs. For example, changes were made to simplify the specification of output control data. This could also be done with a separate program that builds the data file for output control. However, controlling output is such a fundamental part of model operation that it is considered important to have an easy to use capability within the model itself. Another example is the ability to write computed head in formatted files. MODFLOW-88 can write head in unformatted (binary) files; this capability was originally included for the purpose of making it possible for supplemental programs to use head for further analysis. Although unformatted files are the most compact way to store numbers with high precision, many programs that one might want to use to process model-calculated head cannot read the unformatted files. Therefore, an intermediate program to translate MODFLOW's files was required; however, this need to use an intermediate program has tended to discourage the use of supplemental programs. The ability to write formatted head files will make it possible for a wide variety of programs to directly use head files from MODFLOW.

DESCRIPTION OF CHANGES

Highly Visible Changes

The changes in this section will be of interest to any user of MODFLOW. They are primarily changes that affect input and output of data.

Opening Files

Fortran reads and writes files using file units, which are integer numbers. Files are stored on a computer using a file name; thus, there must be a way to associate a file name with a Fortran unit number. As originally programmed, MODFLOW used preconnection to define the connection between file names and units. Preconnection is part of Fortran 77 (ANSI, 1978, p. B-8), and means that the connection between files and units is made externally from Fortran. MODFLOW-96 uses the Fortran OPEN statement to make the connection between a file name and a unit. The user must prepare a special file, termed the "name file", that contains the name and unit of each file to be used. MODFLOW-96 opens all the files listed. Also, the name file contains a file type for each file. The file type specifications are used to indicate which model options are being used rather than specifying model options by reading values for the IUNIT array (McDonald and Harbaugh, 1988, p. 3-25). For example, there is a "RCH" file type. If a file is of type RCH, then this indicates that the RCH (Recharge) Package is being used. The input record that was previously used to specify the IUNIT array has been replaced with a record that is used to specify new options. Unless one of the new options is required, an old data file that contains an IUNIT record can be used without change. The old IUNIT record will be interpreted to mean that none of the new options are to be used. The MAIN program must open the name file. The MAIN program obtains the name of the name file by reading a response to a prompt on the default input device. The name file is opened on unit 99. The method of defining the name file and the unit used can be changed if desired.

Specifying Data Read by the Array-Reading Modules

Several enhancements have been made to the array-reading Utility Modules. Most arrays are read using these modules. The array-reading modules first read a control record that specifies which of three methods will be used to define the array values. The three methods are to make all elements of the array a constant, read all the array elements from the same unit from which the control record is being read, or read all the array elements from a different unit. These options were originally specified using a numeric parameter called "LOCAT," but many users have reported that they have been confused by the numeric LOCAT values. Therefore, new forms of the array control record have been added in which the first field is a word that indicates the method of defining the array. Also, the fields in the new forms of the control record are read using free format rather than fixed format. This means that values do not have to be in specific columns within the record. Instead, values are separated by one or more blanks, a single comma, or a single comma plus any number of blanks on either side of the comma. If no recognized word is found in the first field of a control record, then the record is read using the original fixed format. Thus existing data files need not be changed.

An additional method of defining where array values are read has been added. Instead of specifying the file unit, which refers to a file in the name file, a file name can be supplied directly on the control record. The array reader opens this file on a predefined file unit, reads the array, and closes the file.

The ability to read array values with a free format has also been added. Although the format for reading array values has always been specified in the control record, the format had to be a fixed format. In the modified model, a format of "(FREE)" indicates that free format should be used for reading the array values.

Specifying Data That Are Not Read By The Array-Reading Modules

In MODFLOW-83 and MODFLOW-88, data are read exclusively using fixed formats. If values are not entered in exactly the right column locations, the values will be incorrectly read. This has been a frequent cause of errors. As previously mentioned, provision for reading array data using free format has been added to MODFLOW-96. A further option has been added to MODFLOW-96 that causes all other numeric values to be read using free format. When this option is used, data values can be anywhere in a record provided that they are in the specified order and that each value is separated by a comma with optional blanks, or by one or more blanks. If the free-format option is not used, data files that use the original fixed formats will be correctly read. Details of the requirements for specifying free-format data are contained in the Input Instructions section.

Listing File

Although the MODFLOW documentation frequently refers to printed output, this actually means model output in text form that might possibly be sent to a printer. Such output is actually written to a file called the listing file prior to the actual printing.

MODFLOW was originally designed so that lines in the listing file were as much as 132 characters wide because the standard computer printers that were in use at the time were designed for this width. These printers printed 10 characters per inch on paper that was 14 inches wide. Today, many computer printers use standard 8.5-inch wide paper. Such printers either cannot print 132 columns per line or printing 132 columns results in very small print that is difficult to read. Further, the listing file is frequently examined on a video display screen instead of printing it or in advance of printing it. Many display screens do not support 132 columns, or again if 132 columns can be displayed, the characters are reduced to a small size. Therefore, MODFLOW-96 has been modified to make it possible to limit the length of all lines in the listing file to 80 characters or less.

The smaller limit on the number of characters in a line means that the listing file looks different than it does for MODFLOW-88. Output lines were reduced in length by removing spaces, using more condensed wording, or using additional lines. New user selectable formats have been added for printing arrays, such as head and drawdown, using 80 columns or less; however, the original wide formats for printing arrays are also still included in MODFLOW-96. Thus, array output will continue to be wider than 80 characters unless one of the new narrow formats is specified. (Formats for writing head and drawdown in the listing file are specified in the Output Control Option -- variables IHEDFM and IDDNFM. Formats for writing arrays into the listing file after they have been read by MODFLOW are specified in the array control records -- variable IPRN.)

In addition to changing the width of the overall volumetric budget table, the values in the budget are now written as decimal numbers without exponents and with aligned decimals over a large range of magnitudes. In MODFLOW-88 budget values were written without exponents for only a small range of values, and the decimals were not aligned. The aligned decimal points make it easier to visually compare the magnitude of values for the various budget terms. A disadvantage of the new budget formatting is that more digits may be printed than are significant. The number of digits printed is computer dependent, but users should be aware that on a computer with 32-bit precision for single precision real numbers, there will be only 6 to 7 significant digits.

Greater Flexibility in Generating Files for Interfacing with Other Programs

Two enhancements have been made to MODFLOW-96 in order to make it easier for other programs to use results from MODFLOW.

1. An option to save head and drawdown in formatted files has been added. Previously, head and drawdown could only be saved in unformatted files. The new option avoids the need for using data translation programs in many situations.

2. The ability to save budget data for individual cells has been expanded to include more compact methods of storage. The method of storing model-calculated budget data in MODFLOW-88 is sometimes wasteful of disk space depending on which budget term is being stored; for each budget term saved, one value is saved for each cell in the model. Many of the budget terms are zero at most cells in a typical simulation (for example flow to rivers or flow to wells), so disk space can be saved by storing only the nonzero values. Three new utility modules were added to store budget data. The Output Control Option was enhanced to allow the user to indicate that compact budget files are desired. When this option is specified, each package's budget module calls the utility module that will result in the most compact budget file. The original method can also be used, which ensures that existing applications that read the original budget files will still work without change. The new method of saving budget data will be particularly useful when many budget terms must be saved for many time steps as occurs when tracking particles in a transient simulation.

Output Control

The Output Control Option was included in MODFLOW in order to provide flexibility in obtaining model output (McDonald and Harbaugh, 1988, p. 3-24). The user specifies a number of coded parameters in order to control the output. Many users have reported that the specification of the coded parameters is confusing. Accordingly, a new method for specifying output control information using words has been added to MODFLOW. The new method is easier to understand, and it usually requires less input data. The original method of specifying output control parameters still works also.

Cross-Sectional Models

MODFLOW was developed to simulate ground-water flow in three dimensions. Using the finite-difference method, the domain in which flow is to be simulated is divided into a rectilinear mesh of rows, columns, and layers. For typical three-dimensional problems, it is convenient to enter much of the input data by model layer, and this is the way the input of data is organized in MODFLOW-88. Similarly, the computed heads are output by model layer.

MODFLOW can also be used to simulate two-dimensional flow. One type of two-dimensional flow system is a cross section in which there is no flow perpendicular to the section. Such a system can be simulated in MODFLOW as a single row with multiple layers and columns, but it is cumbersome to use MODFLOW-88 in this way because of the focus on organizing data by layers for input and output. Specifically, calculated head and drawdown are printed as a series of arrays, one for each layer. Each array thus is only one row of data, yet each array starts on a new printout page. Likewise, initial head and the boundary array, IBOUND, are read as a series of two-dimensional arrays. This requires a control record for each one-row layer. An option has been added to MODFLOW-96 to facilitate input and output of data for cross sections. This option causes head to be printed as a single 2-dimensional array (dimensioned by the number of columns and layers) for the cross section. In addition, initial head and IBOUND are read and printed as single arrays. All other data arrays are read and printed by layer as before. These changes only affect the way data are read and printed. Internally, head and IBOUND are stored and processed as 3-dimensional arrays as before. The maximum number of layers was also increased from 80 to 200, which makes it possible to make larger cross sections.

Less Visible Changes

A variety of other changes were made in MODFLOW-96. The casual model user will probably not be interested in this section. Many of these changes will be of interest only to advanced model users that know the Fortran programming language.

Precision of Budget Calculations

In a digital computer, numbers are calculated and stored using a finite number of significant digits. Fortran 77 supports two levels of precision, single and double. The specific precision of both levels is computer dependent, but on most computers single precision has about 6.5 decimal digits of precision, and double precision has about 14 decimal digits of precision. Double precision requires greater memory and computational time, so care was taken when MODFLOW was designed to use minimal double precision while still being able to simulate a wide variety of problems (McDonald and Harbaugh, 1988, p. A-2). Budget calculations are done entirely in single precision in the original model, and in two situations significant budget errors can occur as a result of the single-precision budget calculations.

The first situation for which significant budget errors can occur in the original model is when the model grid has approximately 100,000 or more cells. The total flow for a budget term is calculated by summing the individual flows from each model cell. When a small number is added to a much larger number, part of the small number is lost; and if thousands of small parts are lost, the total may be noticeably smaller than it should be. The other situation in which budget errors can occur in the original model involves the head-dependent stresses simulated using the River, Drain, Evapotranspiration, and General-Head Boundary Packages. All of these packages calculate a flow that is the product of a constant conductance parameter times a head difference. Although in the original model head is calculated as a double precision value, head is converted to single precision prior to calculating budget flows in the head-dependent stress packages. The use of the less accurate value for head causes a small error in the calculated head difference. When the conductance is very large, a small error in head difference can cause a large error in the calculated flow. Both of these errors can be reduced by using double precision in the budget calculations. In the modified model, the budget flows within each package are accumulated using double precision, and head difference in the head-dependent budget packages is calculated using double precision.

Although the use of double precision in budget calculations can avoid inaccuracies in some situations, sometimes additional use of double precision is necessary. That is, precision problems may affect more than just the budget calculations. For example, in large models, it may not be possible to solve the flow equations unless all of the arrays used in the solution algorithm are converted to double precision. Another example is when a high conductance head-dependent stress occurs in a cell along with another stress. All stress terms in the flow equation are accumulated in the single precision RHS and HCOF arrays, and these single precision arrays may not be able to accurately accumulate all terms if they differ significantly in magnitude. Further, when high-conductance head-dependent stresses are used, a much more accurate solution for head is required than usual in order to accurately define the flow. In these situations, it may be necessary to fully convert the model to double precision. The process of doing this has been simplified slightly. There are two steps rather than the 3 steps for MODFLOW-88 (McDonald and Harbaugh, 1988, p. A-3):

1. Add the statement

```
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
```

to the beginning of the specifications section of <u>every</u> module (subroutine) in the model.

2. In Module BAS5AL, change the statement

```
ISUM=ISUM+2*NRCL
```

to

```
ISUM=ISUM+NRCL
```

Note, however, that packages not specifically converted to MODFLOW-96 standards might require additional conversion steps.

Ability to Add Simulation Capabilities

MODFLOW was designed with a modular structure partly for the purpose of facilitating the addition of capabilities. A number of changes have been made to further enhance the ability to add new functionality.

1. Budget modules for all packages have been changed in order to allow access to the calculated budget data by other modules. This would be useful, for example, if a transport capability were added to MODFLOW. Each budget module has been changed to store its budget terms in the BUFF array regardless of whether the budget terms are being saved in a disk file.

Most packages calculate only one budget term, so the values can be readily stored in the 3-dimensional BUFF array. The BCF budget module calculates five budget terms, so it was necessary to restructure it in order to allow access to these budget terms by other modules. The BCF Package calculates flow to storage in transient simulations, flow to constant-head cells, and three terms for flow to adjacent cells. The single BCF budget module of MODFLOW-88 has been replaced by three submodules. The flow to adjacent cell calculations are now done within a new submodule that is called three times from the main program. Each call calculates the values for flow in one direction and puts these values in the BUFF array. The constant-head and storage budget terms have been placed in separate submodules that are each called from the main program. Each submodule returns its values in the BUFF array.

2. An additional method of returning budget values for use by other modules has been provided for the RIV, DRN, WEL, and GHB Packages. These packages maintain a list of data values for each stress. For example, in the WEL Package, the list contains the layer, row, column, and recharge rate for each well. An option has been added to include budget flow as an additional data value in the list of data for any of these stresses. The budget modules will store the calculated budget values in the list when this option is specified. Thus, other modules can obtain budget data for one of these stresses from its list of data values.

3. There is no need to calculate flow between adjacent constant-head cells in MODFLOW because this flow has no affect on the model-calculated heads; however, in some situations, such as transport models, it may be useful or necessary to calculate this flow. Thus, the option to calculate flow between adjacent constant-head cells has been added to the BCF Package. This will only affect the budget terms for constant-head and flow between adjacent cells; it will not affect calculated head. When flow is calculated between adjacent constant-head cells, these flows will be accumulated in the overall flow budget.

4. The ability to read supplemental data has been added to the point stress-term packages (RIV, DRN, WEL, and GHB Packages). Input data for these packages consists of an input record for each model cell that contains a stress. If another package required additional data for each stress, it would be awkward to include the additional data in a separate file. The additional file would have to duplicate the locations of all the stresses. A more efficient approach is to append the additional data to the existing input records for each stress. Therefore, the option to specify up to five additional values to be read for each stress has been added to each of the point stress-term packages. MODFLOW will allocate memory for the additional values and read them; however, MODFLOW will not use these values for any of its calculations.

Miscellaneous Changes

A number of coding changes were made in MODFLOW-96 for a variety of reasons. These changes are:

1. Each character string in MODFLOW is now declared as a single-character variable. Character strings were originally stored as arrays made up of 4 characters per array element because character data were stored in numeric variables when using Fortran 66 (McDonald and Harbaugh, 1983). When MODFLOW-83 was converted to Fortran 77 (MODFLOW-88), the arrays containing character data were declared to be the character data type; however, multiple 4-character array elements were still used to represent strings longer than 4 characters so that it would be possible to easily convert the code back to Fortran 66 (McDonald and Harbaugh, 1988, p. A-5). Fortran 77 is widely available at present, so there should no longer be a need to convert

the code to Fortran 66. The benefits of converting each string to a single-character variable are that it is easier to see the values of character strings when looking at the code, and it is easier to change the values if desired. The simulation title was also converted to two 80-character strings rather than one 128-character string. This change was made to facilitate the changes that limit the printed-line length to 80 characters or less; each 80-character string is printed on a separate line.

2. Several occurrences of Fortran intrinsic functions have been changed from the specific form to the generic form, which is a capability added to Fortran 77. Generic functions allow a single function name to have different argument types. The use of specific names meant that if a user changed the code to use all double precision, then the function names had to change (Harbaugh and McDonald, 1988, Appendix A). This will no longer be necessary (except for any packages that others may add to MODFLOW-96 in which the functions are not converted to generic functions).

3. The size of the X array, which stores most model data, is now specified in a PARAMETER statement rather than in an assignment statement. This means that only one statement must be changed to change the size of the X array. Previously two statements had to be changed, and problems could occur if one statement was changed and not the other.

4. The program has been changed to prevent division by 0 when calculating the water budget if there are no flows in the model. Although it is meaningless to run a model with no flows, it is possible that this could occur by mistake. The original model, depending on the computer being used, would generally produce a divide by 0 error and abort, which would make it difficult to find the cause of the error. The change allows the budget to be printed, and the user can then see that there are no flows.

5. The Strongly Implicit Solution Procedure (SIP) does not work if a group of two or more cells in a single row or column is isolated from the rest of the model. In the original model, this results in division by 0, which causes the program to abort with little indication of what the problem is. The SIP Package has been modified to check for division by 0. When this is detected, the program will print the location of the cell and stop. Although this does not fix the problem, the location information may help the user resolve the problem.

6. In MODFLOW-88 the term "saving" is used in two ways. The first usage of "saving" involves initial (or starting) head. Although initial head is primarily required to start the simulation, it is also required if drawdown is to be calculated later in the simulation. Thus, "saving" initial heads was used to mean that the initial heads were kept in memory throughout the simulation. The user can choose to save or not save these heads, and by not saving them, less memory is used. The second usage of "saving" is for writing the model calculated heads or drawdown in a disk file for later use by another program. The revised model continues to use "saving" for writing head and drawdown in a disk file, but "saving" is no longer used to indicate that initial head is kept in memory. The term "kept throughout the simulation" is used instead. For example, the message "STARTING HEAD WILL BE SAVED" has been changed to "INITIAL HEAD WILL BE KEPT THROUGHOUT THE SIMULATION."

Compatibility With Code Added By Others

The overall modular structure and the use of global variables are unchanged in MODFLOW-96 compared to MODFLOW-88, so packages that have been added to MODFLOW-88 should still work without change in MODFLOW-96. Although new functionality has been added to some of the Utility Modules, the calling arguments and fundamental functionality of the original Utility Modules are unchanged. All of the new capabilities that were added to the new versions of the Utility Modules, such as the ability to use a word in the array control records to specify how an array is defined, will be automatically included when an "old" package is added to MODFLOW-96. Other new capabilities, such as free-format input of non-array data, can only be incorporated in a package through modification of that package.

Each module in MODFLOW has a 6-character name. One of the characters is the version number. The version number for all modules in MODFLOW-88 is 1; however, many users have modified the code. If a module is modified, the version number should be changed. To distinguish MODFLOW-96 from MODFLOW-88 and its past derivatives, all modules in MODFLOW-96 have been given a version number of 5. This version number is likely to be higher than the version number of existing modules although there is no way to know this for sure.

There was no attempt to make MODFLOW-96 compatible with MODFLOW-88 at the module level. Therefore, modules from a package that has been used in some variation of MODFLOW-88 should not be intermixed with the corresponding modules of MODFLOW-96 without first checking to make sure that the calling arguments and internal details of the modules are compatible. For example, a user may have modified module WEL1RP in MODFLOW-88 to change the way well data are read and named the modified module WEL2RP. WEL2RP would have to have the same calling arguments as WEL1RP in order for it to work. The calling arguments of WEL5RP are not the same as the arguments for WEL1RP, however, so WEL2RP could not be directly substituted for WEL5RP. To obtain the modified functionality, WEL5RP could be modified or the entire modified MODFLOW-88 WEL Package could be substituted for the MODFLOW-96 WEL Package. Of course if the modified MODFLOW-88 WEL Package were substituted for the MODFLOW-96 WEL Package, the enhancements contained in the MODFLOW-96 WEL Package would not be included.

INPUT INSTRUCTIONS

Input data for packages, the Name File, the Output Control Option, and the Utility Modules are grouped by items, which are numbered for reference. Each item consists of input parameters. If a parameter is represented in the program by a variable of the same name, then the parameter is shown in all upper case. Parameters that are not program variables are capitalized. Each parameter is defined after all the items are listed.

A parameter may be a single value or an array. An array parameter is indicated in an input item by the inclusion of the array dimensions in parentheses after the parameter. For example, "IBOUND(NCOL,NROW)" indicates that IBOUND is an array with dimensions of the number of columns and rows in the model grid. Some arrays are read by array reading Utility Modules, and for these, the Utility Module (U1DREL, U2DREL, or U2DINT) is named along with the array. Module U1DREL is used for reading one-dimensional real arrays, U2DREL is used for reading two-dimensional real arrays, and U2DINT is used for reading two-dimensional integer arrays. Input instructions for the array reading Utility Modules are located after the input instructions for all the packages.

The input data for each item must start on a new record. (A record is a line in a file.) It is assumed that all parameters for an item will be contained in a single record unless the item is an array. That is, an array may occupy multiple records. When reading a two-dimensional array that consists of a series of rows of data, each row must start on a new record.

Each parameter has a data type, which can be Real, Integer, or Character. Integers are whole numbers, and these values must not include a decimal point or exponent. Real numbers can include a decimal point and an exponent. If no decimal point is included in the entered value, then the decimal point is assumed to be at the right side of the value. Any printable character is allowed for character parameters. All parameters starting with the letters I-N are integers. Parameters starting with the letters A-H and O-Z are either real numbers or character data, and most of these are real numbers. That is, there are few character parameters in MODFLOW. It should be assumed that a parameter starting with A-H or O-Z is a real number unless otherwise stated in the parameter definitions.

Both free and fixed formatting are used as specified throughout the input instructions. With fixed format, a value has a specified location, called the field, within a record. If a number does not require the entire field width, the number should be right justified within the field, and blanks should fill the remaining width of the field. The justification of text within a field does not matter. Free format is similar to Fortran's list-directed input. With free format, the spacing of values is not fixed. Each value can consist of any number of characters. One or more spaces, or a single comma optionally combined with spaces, must separate adjacent values. A value of 0 must be explicitly represented with 0 and not by one or more spaces because there is no way to detect the difference between a space that represents 0 and a space that represents a value separator.

Two capabilities included in Fortran's list-directed input are not included. First, null values in which input values are left unchanged from their previous values are not allowed. In general, MODFLOW's input values are not defined prior to their input. Second, a "/" cannot be used to terminate an input record without including values for all parameters; data values for all input items must be explicitly specified on an input record. For character data, MODFLOW's free format implementation is less stringent than the list-directed input of Fortran 77. Fortran requires character data to be delineated by apostrophes. MODFLOW does not require apostrophes unless a blank or a comma is part of an array format. (See the Input Instructions for Array Reading Utility Modules section.)

Input Instructions for the Name File

The name file is a special file that is read by the Basic Package. The name file is read on unit 99, as specified in the MAIN program. The name file must specify the names of all files being used in a model simulation except for OPEN/CLOSE files (see Input Instructions for Array Reading Utility Modules). As a minimum, there must be a listing file to which all model output intended for printing will be written and one input file for each package being used. (Output that goes to the listing file is frequently referred to as printed output because it is expected that all or parts of this file will be printed.) There may also be additional input files containing array data and output files for saving head, drawdown, and cell-by-cell budget data.

FOR EACH SIMULATION

```
1.  Ftype     Nunit     Fname
```

The name file must contain one of the above records (item 1) for each file used in the simulation. The first record must be for the listing file; the remaining records can be in any order. All parameters are free format. The length of each record must be 79 characters or less.

Any number of comment records, which are indicated by the # character in column one, can be interspersed among the item-1 records. Any text characters can follow the # character. Comment records have no effect on the simulation; their purpose is to allow users to provide documentation about a particular simulation. All comment records that come after the first item-1 record are written in the listing file.

Explanation of Parameters in the Name File

Ftype--is the file type, which must be one of the following character values. Ftype may be entered in uppercase or lowercase:

LIST for the listing file
BAS for the Basic Package
OC for the Output Control Option
BCF for the Block-Centered Flow Package
RCH for the Recharge Package
RIV for the River Package
WEL for the Well Package
DRN for the Drain Package
GHB for the General-Head Boundary Package
EVT for the Evapotranspiration Package
SIP for the Strongly Implicit Procedure Package
SOR for the Slice-Successive Over-Relaxation Package
DATA(BINARY) for binary (unformatted) files such as those used to save cell-by-cell budget data and binary (unformatted) head and drawdown data.
DATA for formatted (text) files such as those used to save formatted head and drawdown and for input of array data from files that are separate from the primary package input files.

Nunit--is the Fortran unit to be used when reading from or writing to the file. Any legal unit number on the computer being used can be specified except units 97-99. Unit 99 is used for the name file and for reading arrays using the OPEN/CLOSE option (see Input Instructions for Array Reading Utility Modules section). Units 97 and 98 are used for batch files as explained below. Each file must have a unique unit number.

Fname--is the name of the file, which is a character value.

Except as described below, MODFLOW will prompt the user to enter the name of the name file when MODFLOW is started. Once a name file is entered, MODFLOW will run without further user interaction.

Provision has been made to allow a series of simulations to be run without the need to manually restart MODFLOW for each simulation and respond to the prompt for the name file. This is called batch mode and is done through the use of a special file named modflow.bf. If this file exists in the user's current directory when MODFLOW is invoked, the user is not prompted for a name file. Instead, MODFLOW reads the names of name files from modflow.bf. Each line of the file contains the name of a single name file, and each name can be up to 80 characters long.

When batch mode is used, a report file, named modbatch.rpt, is created. Before starting a simulation, the name of the name file is written to the report file. If the name file exists, the simulation is executed; when done, another message is written to the report file. If a name file does not exist, a message is written to the report file and processing continues with the next name file in modflow.bf.

When batch mode is used to run MODFLOW, two additional file units are used -- 97 and 98. These units cannot be used by any of the MODFLOW simulations. However, the file units used by one simulation can be the same as those used in any of the other simulations that are executed because each simulation is executed separately.

Basic Package Input Instructions

Input for the Basic (BAS) Package except for output control input is read from the file that is specified with "BAS" as the file type. Non-array numeric parameters are free format if the word FREE is specified in item 4; otherwise, the non-array parameters all have 10-character fields.

FOR EACH SIMULATION

```
1.  HEADNG     (80 text characters)
2.  HEADNG     (80 text characters)
3.  NLAY    NROW    NCOL    NPER    ITMUNI
4.  Options    (79 text characters)
5.  IAPART  ISTRT
6.  IBOUND(NCOL,NROW) or (NCOL,NLAY) -- U2DINT
```

(If not a cross section, a NCOL by NROW array is read for each layer in the grid.)
(If a cross section, a single NCOL by NLAY array is read for the cross section.)

```
7.  HNOFLO
8.  Shead(NCOL,NROW) or (NCOL,NLAY) -- U2DREL
```

(If not a cross section, an NCOL by NROW array is read for each layer in the grid.)
(If a cross section, a single NCOL by NLAY array is read for the cross section.)

FOR EACH STRESS PERIOD

```
9.  PERLEN   NSTP  TSMULT
```

Explanation of Parameters Read by the BAS Package

HEADNG--is the simulation title that is written to the listing file. HEADNG consists of two 80-character values that are read from two records. Both records must be included even if they are blank.

NLAY--is the number of layers in the model grid. This value is limited to 200 layers.

NROW--is the number of rows in the model grid.

NCOL--is the number of columns in the model grid.

NPER--is the number of stress periods in the simulation.

ITMUNI--indicates the time unit of model data. It is used only for printout of elapsed simulation time. It does not affect model calculations.

```
0 - undefined          3 - hours
1 - seconds            4 - days
2 - minutes            5 - years
```

The unit of time must be consistent for all data values that involve time. For example, if years is the chosen time unit, stress-period length, time-step length, transmissivity, etc., must all be expressed using years for their time units. Likewise, the length unit must also be consistent.

Options--is a character parameter that is scanned for words (separated by one or more spaces) that specify program options. Three options are currently recognized. Unrecognized words are ignored, and a word may be specified in either uppercase or lowercase. A blank record is acceptable and indicates no options. Note that this item was used to specify the IUNIT array in MODFLOW-88 (McDonald and Harbaugh, 1988). Unless a new option is desired, there is no need to modify an old data file because the IUNIT values will be unrecognized words that will be ignored.

XSECTION indicates that the model is a 1-row cross section for which Shead and IBOUND should each be read as single 2-dimensional arrays with dimensions of NCOL and NLAY. Likewise, head and drawdown should be printed and saved in disk files as single 2-dimensional arrays.

CHTOCH indicates that flow between adjacent constant-head cells should be calculated.

FREE indicates that free format is used for input parameters throughout the Basic Package and other packages as indicated in their input instructions. Be sure that all parameters read using free format have a non-blank value and that a comma or at least one blank separates all adjacent values.

IAPART--indicates whether array BUFF is separate from array RHS.
If IAPART = 0, the arrays BUFF and RHS occupy the same space. This option conserves space.
If IAPART ≠ 0, the arrays BUFF and RHS occupy different space.
Generally IAPART should be set to 0 unless another package instructs otherwise. In this report, the only package that sometimes requires IAPART to be nonzero is the BCF Package. Refer to the BCF Package input instructions.

ISTRT--indicates whether initial (starting) heads are to be kept in memory throughout the simulation. They must be kept if drawdown will be calculated during the simulation.
If ISTRT = 0, initial heads are not kept.
If ISTRT ≠ 0, initial heads are kept.

IBOUND--is the boundary array. One value is read for every model cell. Usually, these values are read a layer at a time; however, when the XSECTION option is specified, a single array for the cross section is read. Note that although IBOUND is read as one or more 2-dimensional arrays, it is stored internally as a 3-dimensional array.
If IBOUND(J,I,K) < 0, cell J,I,K has a constant head.
If IBOUND(J,I,K) = 0, cell J,I,K is no flow.
If IBOUND(J,I,K) > 0, cell J,I,K is variable head.

HNOFLO--is the value of head that is assigned to all cells that are initially specified as no-flow cells (IBOUND = 0). Because head at no-flow cells is unused in model calculations, this does not affect model results but serves to identify no-flow cells when head is printed. This value is also substituted for drawdown at all cells that are initially no-flow cells if the drawdown option is used. Even if the user does not anticipate having no-flow cells, a value for HNOFLO must be entered.

Shead--is initial (starting) head -- i.e. head at the beginning of the simulation. All simulations, including steady-state simulations, require initial head to be specified. For steady-state simulations, initial head is used to start the iterative solution process. A value is read for every model cell although initial head is not used for no-flow cells. For constant-head cells, the initial head is the constant head. Usually, these values are read a layer at a time; however, when the XSECTION option is specified, a single array for the cross section is read. Note that although Shead is read as one or more 2-dimensional arrays, it is stored internally as a 3-dimensional array.

PERLEN--is the length of a stress period.

NSTP--is the number of time steps in a stress period.

TSMULT--is the multiplier for the length of successive time steps. The length of a time step is calculated by multiplying the length of the previous time step by TSMULT. The length of the first time step, Δt_1, is related to PERLEN, NSTP, and TSMULT by the relation

$$\Delta t_1 = PERLEN \frac{TSMULT - 1}{TSMULT^{NSTP} - 1}$$

Output Control Option Input Instructions

Input to Output Control is read from the file that is specified as type "OC". If no "OC" file is specified, default output control is used. Under the default, head and overall budget are printed (written to the listing file) at the end of every stress period. Additionally, drawdown is printed if initial heads are kept during the simulation (ISTRT is not 0). The default printout format for head and drawdown is 10G11.4.

There are two ways to specify Output Control data: words and numeric codes. The same method must be used throughout any simulation.

Output Control using words:

Words are shown in bold italics; these words must be entered exactly as shown except that they may be entered in either uppercase or lowercase. Optional parts of records are shown in brackets. One or more spaces must separate each word or parameter, and the total record length must not exceed 79 characters.

FOR EACH SIMULATION

1. Any combination of the following records:

   HEAD PRINT FORMAT IHEDFM

   DRAWDOWN PRINT FORMAT IDDNFM

   HEAD SAVE FORMAT CHEDFM [***LABEL***]

   DRAWDOWN SAVE FORMAT CDDNFM [***LABEL***]

   HEAD SAVE UNIT IHEDUN

   DRAWDOWN SAVE UNIT IDDNUN

   COMPACT BUDGET FILES This indicates that cell-by-cell budget files will require less disk space than files written by MODFLOW-88; however, programs that read this data will have to be modified. If this option is not used, MODFLOW-96 will write the files using the MODFLOW-88 method.

FOR EACH TIME STEP FOR WHICH OUTPUT IS DESIRED

2. ***PERIOD*** IPEROC ***STEP*** ITSOC
3. Any combination of the following records:

   PRINT HEAD [list layers if all layers not desired]

   PRINT DRAWDOWN [list layers if all layers not desired]

   PRINT BUDGET This causes the overall volumetric budget to be written to the listing file.

   SAVE HEAD [list layers if all layers not desired]

   SAVE DRAWDOWN [list layers if all layers not desired]

   SAVE BUDGET This causes cell-by-cell budget data to be written to a file.

(Item 2 and one or more item-3 records are specified for each time for which output is desired. These records must be in the order of increasing simulation time.)

Output Control using numeric codes:

All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the parameters all have 10-character fields.

FOR EACH SIMULATION

1.	IHEDFM	IDDNFM	IHEDUN	IDDNUN

FOR EACH TIME STEP

2.	INCODE	IHDDFL	IBUDFL	ICBCFL
3.	Hdpr	Ddpr	Hdsv	Ddsv

(Item 3 is read 0, 1, or NLAY times, depending on the value of INCODE.)

Explanation of Parameters Read by the Output Control Option

IHEDFM--is a code for the format in which heads will be printed. (Positive values for wrap format; negative values for strip format.)

```
0 - (10G11.4)      10 - (20F5.3)
1 - (11G10.3)      11 - (20F5.4)
2 - (9G13.6)       12 - (10G11.4)
3 - (15F7.1)       13 - (10F6.0)
4 - (15F7.2)       14 - (10F6.1)
5 - (15F7.3)       15 - (10F6.2)
6 - (15F7.4)       16 - (10F6.3)
7 - (20F5.0)       17 - (10F6.4)
8 - (20F5.1)       18 - (10F6.5)
9 - (20F5.2)
```

IDDNFM--is a code for the format in which drawdowns will be printed. The codes are the same as for IHEDFM.

CHEDFM--is a character value that specifies the format for saving heads, and can only be specified if the word method of output control is used. The format must contain 20 characters or less and must be a valid Fortran format that is enclosed in parentheses. The format must be enclosed in apostrophes if it contains one or more blanks or commas. The optional word ***LABEL*** after the format is used to indicate that each layer of output should be preceded with a line that defines the output (simulation time, the layer being output, etc.). If there is no record specifying CHEDFM, then heads are written to a binary (unformatted) file.

CDDNFM--is a character value that specifies the format for saving drawdown, and can only be specified if the word method of output control is used. The format must contain 20 characters or less and must be a valid Fortran format that is enclosed in parentheses. The format must be enclosed in apostrophes if it contains one or more blanks or commas. The optional word ***LABEL*** is used to indicate that each layer of output should be preceded with a line that defines the output (simulation time, the layer being output, etc.). If there is no record specifying CDDNFM, then drawdown is written to a binary (unformatted) file.

IHEDUN--is the unit number on which heads will be saved.

IDDNUN--is the unit number on which drawdowns will be saved.

IPEROC--is the stress period number at which output is desired.

ITSOC--is the time step number (within a stress period) at which output is desired.

INCODE--is the code for reading item 3.
  If INCODE < 0, item 3 flags are used from the last time step. Item 3 is not read.
  If INCODE = 0, all layers are treated the same way. Item 3 will consist of one record.
  If INCODE > 0, item 3 will consist of one record for each layer.

IHDDFL--is a head and drawdown output flag. This flag allows item 3 flags to be specified in an early time step and then used or not used in subsequent time steps. Thus, it may be possible to use IHDDFL to avoid resetting item 3 flags every time step.
  If IHDDFL = 0, no heads or drawdowns will be printed or saved regardless of which item 3 flags are specified.
  If IHDDFL ≠ 0, heads and drawdowns will be printed or saved according to the item 3 flags.

IBUDFL--is a budget print flag.
  If IBUDFL = 0, overall volumetric budget will not be printed.
  If IBUDFL ≠ 0, overall volumetric budget will be printed.

ICBCFL--is a cell-by-cell flow-term flag.
  If ICBCFL = 0, cell-by-cell flow terms are not saved or printed.
  If ICBCFL ≠ 0, cell-by-cell flow terms are printed or recorded on disk depending on flags set in the component of flow packages, i.e., IWELCB, IRCHCB, etc.

Hdpr--is the output flag for head printout.
  If Hdpr = 0, head is not printed for the corresponding layer.
  If Hdpr ≠ 0, head is printed for the corresponding layer.

Ddpr--is the output flag for drawdown printout.
  If Ddpr = 0, drawdown is not printed for the corresponding layer.
  If Ddpr ≠ 0, drawdown is printed for the corresponding layer.

Hdsv--is the output flag for head save.
  If Hdsv = 0, head is not saved for the corresponding layer.
  If Hdsv ≠ 0, head is saved for the corresponding layer.

Ddsv--is the output flag for drawdown save.
  If Ddsv = 0, drawdown is not saved for the corresponding layer.
  If Ddsv ≠ 0, drawdown is saved for the corresponding layer.

Example Output Control Input Using Words:

(Note that the first line cannot be blank, but any subsequent blank lines after the first line are ignored when the word method is used to specify Output Control data. Indented lines are allowed because of the use of free format input.)

```
HEAD PRINT FORMAT 15
HEAD SAVE FORMAT (20F10.3) LABEL
HEAD SAVE UNIT 30
COMPACT BUDGET FILES
DRAWDOWN PRINT FORMAT 14

PERIOD 1 STEP 1
   PRINT HEAD 2 6
   PRINT DRAWDOWN
   PRINT BUDGET
   SAVE BUDGET
   SAVE HEAD

PERIOD 1 STEP 7
   SAVE HEAD 1 3 5
   PRINT DRAWDOWN
   SAVE BUDGET

PERIOD 2 STEP 5
   PRINT HEAD
   PRINT BUDGET
   SAVE BUDGET
   SAVE HEAD
```

Block-Centered Flow Package Input Instructions

This version of the Block-Centered Flow (BCF) Package combines the capabilities documented in McDonald and Harbaugh (1988), McDonald and others (1992), and Goode and Appel (1992). Input for the BCF Package is read from the file that is type "BCF."

FOR EACH SIMULATION

1. `ISS   IBCFCB   HDRY   IWDFLG   WETFCT   IWETIT   IHDWET`
   These 7 parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the parameters all have 10-character fields.
2. `Ltype(NLAY)`
   These array values are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, they have fixed-format fields. The fixed format fields are each 2 characters wide with 40 values per line. Use only as many lines as required for the number of model layers.
3. `TRPY(NLAY) -- U1DREL`
4. `DELR(NCOL) -- U1DREL`
5. `DELC(NROW) -- U1DREL`

A subset of the following two-dimensional arrays are used to describe each layer. The arrays needed for each layer depend on the layer-type code (LAYCON, which is defined as part of the Item-2 Ltype array), whether the simulation is transient (ISS = 0) or steady state (ISS not 0), and if the wetting capability is active (IWDFLG not 0). If an array is not needed, it must be omitted. In no situation will all arrays be required. The required arrays (items 6-13) for layer 1 are read first; then the arrays for layer 2, etc.

6. `Sf1(NCOL,NROW)-- U2DREL` If transient (ISS=0).
7. `Tran(NCOL,NROW) -- U2DREL` If LAYCON (see Ltype array) IS 0 or 2.
8. `HY(NCOL,NROW) -- U2DREL` If LAYCON (see Ltype array) is 1 or 3.
9. `BOT(NCOL,NROW) -- U2DREL` If LAYCON (see Ltype array) is 1 or 3.
10. `Vcont(NCOL,NROW) -- U2DREL` If not the bottom layer.
11. `Sf2(NCOL,NROW) -- U2DREL` If transient (ISS=0) and
    LAYCON (see Ltype array) is 2 or 3.
12. `TOP(NCOL,NROW) -- U2DREL` If LAYCON (see Ltype array) is 2 or 3.
13. `WETDRY(NCOL,NROW) -- U2DREL` If IWDFLG is not 0 and
    LAYCON (see Ltype array) is 1 or 3.

Explanation of Parameters Read by the BCF Package

ISS--is the steady-state flag.
If ISS ≠ 0, the simulation is steady state.
If ISS = 0, the simulation is transient.

IBCFCB--is a flag and a unit number.
If IBCFCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control. The terms that are saved are storage, constant-head flow, and flow between adjacent cells.
If IBCFCB = 0, cell-by-cell flow terms will not be written.
If IBCFCB < 0, cell-by-cell flow for constant-head cells will be written in the listing file when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control. Cell-by-cell flow to storage and between adjacent cells will not be written to any file.

HDRY--is the head that is assigned to cells that are converted to dry during a simulation. Although this value plays no role in the model calculations, it is useful as an indicator when looking at the resulting heads that are output from the model. HDRY is thus similar to HNOFLO in the Basic Package, which is the value assigned to cells that are no-flow cells at the start of a model simulation.

IWDFLG--is a flag that determines if the wetting capability is active.
If IWDFLG = 0, the wetting capability is inactive.
If IWDFLG is not 0, the wetting capability is active.

WETFCT--is a factor that is included in the calculation of the head that is initially established at a cell when it is converted from dry to wet. (See IHDWET.)

IWETIT--is the iteration interval for attempting to wet cells. Wetting is attempted every IWETIT iterations. If using the PCG solver (Hill, 1990), this applies to outer iterations, not inner iterations. If IWETIT is 0, it is changed to 1.

IHDWET--is a flag that determines which equation is used to define the initial head at cells that become wet:
If IHDWET = 0, equation (3a) (McDonald and others, 1992) is used:
$h = BOT + WETFCT (h_n - BOT)$
If IHDWET is not 0, equation (3b) (McDonald and others, 1992) is used:
$h = BOT + WETFCT (WETDRY)$

Ltype--is a combined code specifying the layer type (LAYCON) and the method of computing interblock conductance. Read one value for each layer; each element holds the code for the respective layer. There is a limit of 200 layers. Use as many records as needed to enter a value for each layer. Values are two-digit numbers. The left digit defines the method of calculating interblock transmissivity. The methods are described by Goode and Appel (1992).
0 or blank - harmonic mean (the method used in MODFLOW-88).
1 - arithmetic mean
2 - logarithmic mean
3 - arithmetic mean of saturated thickness and logarithmic mean of transmissivity. RHS array must be separate from BUFF array in memory (IAPART must be nonzero in BAS Package input).

The right digit defines the layer type (LAYCON), which is the same as in MODFLOW-88:

0 - confined -- Transmissivity and storage coefficient of the layer are constant for the entire simulation.

1 - unconfined -- Transmissivity of the layer varies. It is calculated from the saturated thickness and hydraulic conductivity. The storage coefficient is constant. This type code is valid only for layer 1.

2 - confined/unconfined -- Transmissivity of the layer is constant. The storage coefficient may alternate between confined and unconfined values. Vertical flow from above is limited if the layer desaturates.

3 - confined/unconfined -- Transmissivity of the layer varies. It is calculated from the saturated thickness and hydraulic conductivity. The storage coefficient may alternate between confined and unconfined values. Vertical flow from above is limited if the aquifer desaturates.

TRPY--is a one-dimensional array containing an anisotropy factor for each layer. It is the ratio of transmissivity or hydraulic conductivity (whichever is being used) along a column to transmissivity or hydraulic conductivity along a row. Set to 1.0 for isotropic conditions. This is a single array with one value per layer. Do not read an array for each layer; include only one array control record for the entire array.

DELR--is the cell width along rows. Read one value for each of the NCOL columns. This is a single array with one value for each column.

DELC--is the cell width along columns. Read one value for each of the NROW rows. This is a single array with one value for each row.

Sf1--is the primary storage coefficient. Read only for a transient simulation (steady-state flag, ISS, is 0). For LAYCON equal to 1, Sf1 will always be specific yield, while for LAYCON equal to 2 or 3, Sf1 will always be confined storage coefficient. For LAYCON equal to 0, Sf1 would normally be confined storage coefficient; however, a LAYCON value of 0 can also be used to simulate water-table conditions where drawdowns are expected to remain everywhere a small fraction of the saturated thickness, and where there is no layer above, or flow from above is negligible. In this case, specific yield values would be entered for Sf1.

Tran--is the transmissivity along rows. Tran is multiplied by TRPY to obtain transmissivity along columns. Read only for layers where LAYCON is 0 or 2.

HY--is the hydraulic conductivity along rows. HY is multiplied by TRPY to obtain hydraulic conductivity along columns. Read only for layers where LAYCON is 1 or 3.

BOT--is the elevation of the aquifer bottom. Read only for layers where LAYCON is 1 or 3.

Vcont--is the vertical hydraulic conductivity divided by the thickness from a layer to the layer below. The value for a cell is the hydraulic conductivity divided by thickness for the material between the node in that cell and the node in the cell below. Because there is not a layer beneath the bottom layer, Vcont cannot be specified for the bottom layer.

Sf2--is the secondary storage coefficient. Read only for layers where LAYCON is 2 or 3 and only if the simulation is transient (steady-state flag, ISS, is 0). The secondary storage coefficient is always specific yield.

TOP--is the elevation of the aquifer top. Read only for layers where LAYCON is 2 or 3.

WETDRY--is a combination of the wetting threshold and a flag to indicate which neighboring cells can cause a cell to become wet. If WETDRY < 0, only the cell below a dry cell can cause the cell to become wet. If WETDRY > 0, the cell below a dry cell and the four horizontally adjacent cells can cause a cell to become wet. If WETDRY is 0, the cell cannot be wetted. The absolute value of WETDRY is the wetting threshold. When the sum of BOT and the absolute value of WETDRY at a dry cell is equaled or exceeded by the head at an adjacent cell, the cell is wetted. Read only if LAYCON is 1 or 3 and IWDFLG is not 0.

River Package Input Instructions

Input to the River (RIV) Package is read from the file that has file type "RIV." Optional parameters are shown in brackets. All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the non-optional parameters have 10-character fields and the optional parameters are free format.

FOR EACH SIMULATION

```
1.  MXRIVR  IRIVCB  [Option]
```

The total record length must not exceed 79 characters.

FOR EACH STRESS PERIOD

```
2.  ITMP
3.  Layer   Row   Column   Stage   Cond   Rbot   [xyz]
```

(Item 3 normally consists of one record for each river reach. If ITMP is negative or zero, item 3 is not read. The total record length must not exceed 150 characters.)

Explanation of Parameters Read by the RIV Package

MXRIVR--is the maximum number of river reaches active in any stress period.

IRIVCB--is a flag and a unit number.

If IRIVCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

If IRIVCB = 0, cell-by-cell flow terms will not be written.

If IRIVCB < 0, river leakage for each reach will be written to the listing file when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

Option--is an optional list of character values.

AUXILIARY abc or AUX abc -- "abc" is the name of an auxiliary parameter to be read for each river reach as part of item 3. Up to 5 parameters can be specified, each of which must be preceded by "AUXILIARY" or "AUX." These parameters will not be used by the RIV Package, but they will be available for use by other packages. The auxiliary parameter values will be read after the Rbot parameter.

CBCALLOCATE or CBC -- indicates that memory should be allocated to store cell-by-cell flow for each river reach in order to make these flows available for use in other packages.

ITMP--is a flag and a counter.

If ITMP < 0, river data from the last stress period will be reused.

If ITMP ≥ 0, ITMP will be the number of reaches active during the current stress period.

Layer--is the layer number of the cell containing the river reach.

Row--is the row number of the cell containing the river reach.

Column--is the column number of the cell containing the river reach.

Stage--is the head in the river.

Cond--is the riverbed hydraulic conductance.

Rbot--is the elevation of the bottom of the riverbed.

[xyz]--is up to 5 auxiliary parameters for a river reach that have been defined in item 1. The auxiliary parameters must be present in each item-3 record if they are defined in item 1.

Recharge Package Input Instructions

Input to the Recharge (RCH) Package is read from the file that has type "RCH." All non-array parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the non-array parameters have 10-character fields.

```
FOR EACH SIMULATION
     1.   NRCHOP    IRCHCB

FOR EACH STRESS PERIOD
     2.   INRECH    INIRCH
     3.   RECH(NCOL,NROW) -- U2DREL   If INRECH ≥0
     4.   IRCH(NCOL,NROW) -- U2DINT   If NRCHOP=2 and if INIRCH≥0
```

Explanation of Parameters Read by the RCH Package

NRCHOP--is the recharge option code. Recharge fluxes are defined in a two-dimensional array, RECH, with one value for each vertical column. Accordingly, recharge is applied to one cell in each vertical column, and the option code determines which cell in the column is selected for recharge.

1 - Recharge is only to the top grid layer.
2 - Vertical distribution of recharge is specified in array IRCH.
3 - Recharge is applied to the highest active cell in each vertical column. A constant-head node intercepts recharge and prevents deeper infiltration.

IRCHCB--is a flag and a unit number.

If IRCHCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.
If IRCHCB ≤ 0, cell-by-cell flow terms will not be written.

INRECH--is the RECH read flag.

If INRECH ≥ 0, an array of recharge fluxes, RECH, is read.
If INRECH < 0, recharge rates from the preceding stress period are used.

INIRCH--is the IRCH read flag, which is read only if NRCHOP is two:

If INIRCH ≥ 0, an array of layer numbers (IRCH) is read.
If INIRCH < 0, the array (IRCH) used in the preceding stress period is reused.

RECH--is the recharge flux (LT^{-1}). Read only if INRECH is greater than or equal to zero.

IRCH--is the layer number array that defines the layer in each vertical column where recharge is applied. Read only if NRCHOP is two and if INIRCH is greater than or equal to zero.

Well Package Input Instructions

Input to the Well (WEL) Package is read from the file that has type "WEL." Optional parameters are shown in brackets. All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the non-optional parameters have 10-character fields and the optional parameters are free format.

FOR EACH SIMULATION

```
1. MXWELL  IWELCB  [Option]
```

The total record length must not exceed 79 characters.

FOR EACH STRESS PERIOD

```
2. ITMP
3. Layer   Row   Column   Q   [xyz]
```

(Item 3 normally consists of one record for each well. If ITMP is negative or zero, item 3 is not read. The total record length must not exceed 150 characters.)

Explanation of Parameters Read by the WEL Package

MXWELL--is the maximum number of wells used in any stress period.

IWELCB--is a flag and a unit number.

If IWELCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

If IWELCB = 0, cell-by-cell flow terms will not be written.

If IWELCB < 0, well recharge for each well will be written to the listing file when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

Option--is an optional list of character values.

AUXILIARY abc or AUX abc -- "abc" is the name of an auxiliary parameter to be read for each well as part of item 3. Up to 5 parameters can be specified, each of which must be preceded by "AUXILIARY" or "AUX." These parameters will not be used by the WEL Package, but they will be available for use by other packages. The auxiliary parameter values will be read after the Q parameter.

CBCALLOCATE or CBC -- indicates that memory should be allocated to store cell-by-cell flow for each well in order to make these flows available for use in other packages.

ITMP--is a flag and a counter.

If ITMP < 0, well data from the last stress period will be reused.

If ITMP ≥ 0, ITMP will be the number of wells active during the current stress period.

Layer--is the layer number of the model cell that contains the well.

Row--is the row number of the model cell that contains the well.

Column--is the column number of the model cell that contains the well.

Q--is the volumetric recharge rate. A positive value indicates recharge and a negative value indicates discharge (pumping).

[xyz]--is up to 5 auxiliary parameters for a well that have been defined in item 1. The auxiliary parameters must be present in each item-3 record if they are defined in item 1.

Drain Package Input Instructions

Input to the Drain (DRN) Package is read from the file that has type "DRN." Optional parameters are shown in brackets. All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the non-optional parameters have 10-character fields and the optional parameters are free format.

FOR EACH SIMULATION

```
1.  MXDRN    IDRNCB   [Option]
```

The total record length must not exceed 79 characters.

FOR EACH STRESS PERIOD

```
2.  ITMP
3.  Layer    Row    Column    Elevation    Cond    [xyz]
```

(Item 3 normally consists of one record for each drain. If ITMP is negative or zero, item 3 will not be read. The total record length must not exceed 150 characters.)

Explanation of Parameters Read by the DRN Package

MXDRN--is the maximum number of drain cells active in any stress period.

IDRNCB--is a flag and a unit number.

If IDRNCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

If IDRNCB = 0, cell-by-cell flow terms will not be written.

If IDRNCB < 0, drain leakage for each drain cell will be written to the listing file when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

Option--is an optional list of character values.

AUXILIARY abc or AUX abc -- "abc" is the name of an auxiliary parameter to be read for each drain as part of item 3. Up to 5 parameters can be specified, each of which must be preceded by "AUXILIARY" or "AUX." These parameters will not be used by the DRN Package, but they will be available for use by other packages. The auxiliary parameter values will be read after the Cond parameter.

CBCALLOCATE or CBC -- indicates that memory should be allocated to store cell-by-cell flow for each drain in order to make these flows available for use in other packages.

ITMP--is a flag and a counter.

If ITMP < 0, drain data from the last stress period will be reused.

If ITMP ≥ 0, ITMP will be the number of drains active during the current stress period.

Layer--is the layer number of the cell containing the drain.

Row--is the row number of the cell containing the drain.

Column--is the column number of the cell containing the drain.

Elevation--is the elevation of the drain.

Cond--is the hydraulic conductance of the interface between the aquifer and the drain.

[xyz]--is up to 5 auxiliary parameters for a drain that have been defined in item 1. The auxiliary parameters must be present in each item-3 record if they are defined in item 1.

Evapotranspiration Package Input Instructions

Input to the Evapotranspiration (EVT) Package is read from the file that is type "EVT." All non-array parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the non-array parameters have 10-character fields.

FOR EACH SIMULATION

```
1.  NEVTOP     IEVTCB
```

FOR EACH STRESS PERIOD

```
2.  INSURF      INEVTR      INEXDP     INIEVT
3.  SURF(NCOL,NROW) -- U2DREL  If INSURF ≥ 0
4.  EVTR(NCOL,NROW) -- U2DREL  If INEVTR ≥ 0
5.  EXDP(NCOL,NROW) -- U2DREL  If INEXDP ≥ 0
6.  IEVT(NCOL,NROW) -- U2DINT  If NEVTOP=2 and if INIEVT ≥ 0
```

Explanation of Parameters Read by the EVT Package

NEVTOP--is the evapotranspiration (ET) option code. ET parameters (ET surface, maximum ET rate, and extinction depth) are specified in two-dimensional arrays, SURF, EVTR, and EXDP, with one value for each vertical column. Accordingly, ET is calculated for one cell in each vertical column. The option codes determine for which cell in the column ET will be calculated.

1 - ET is calculated only for cells in the top grid layer.
2 - The cell for each vertical column is specified by the user in array IEVT.

IEVTCB--is a flag and a unit number.

If IEVTCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.
If IEVTCB ≤ 0, cell-by-cell flow terms will not be written.

INSURF--is the ET surface (SURF) read flag.

If INSURF ≥ 0, an array containing the ET surface elevation (SURF) will be read.
If INSURF < 0, the ET surface from the preceding stress period will be reused.

INEVTR--is the maximum ET rate (EVTR) read flag.

If INEVTR ≥ 0, an array containing the maximum ET rate (EVTR) will be read.
If INEVTR < 0, the maximum ET rate from the preceding stress period will be reused.

INEXDP--is the extinction depth (EXDP) read flag.

If INEXDP ≥ 0, an array containing the extinction depth (EXDP) will be read.
If INEXDP < 0, the extinction depth from the preceding stress period will be reused.

INIEVT--is the layer indicator (IEVT) read flag. It is read only if the ET option (NEVTOP) is equal to two.

If INIEVT ≥ 0, an array containing the layer indicators (IEVT) will be read.
If INIEVT < 0, layer indicators used during the preceding stress period will be reused.

SURF--is the elevation of the ET surface. This array is read only if INSURF≥0.

EVTR--is the maximum ET flux (volumetric flow rate per unit area (LT^{-1})). This array is read only if INEVTR≥0.

EXDP--is the ET extinction depth. This array is read only if INEXDP≥0.

IEVT--is the layer indicator array. For each horizontal location, it indicates the layer from which ET is removed. It is read only if the ET option is equal to two and if INIEVT≥0.

General-Head Boundary Package Input Instructions

Input to the General-Head Boundary (GHB) Package is read from the file that is type "GHB." Optional parameters are shown in brackets. All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, the non-optional parameters have 10-character fields and the optional parameters are free format.

FOR EACH SIMULATION

```
1. MXBND    IGHBCB   [Option]
```

The total record length must not exceed 79 characters.

FOR EACH STRESS PERIOD

```
2. ITMP
3. Layer    Row    Column    Bhead    Cond    [xyz]
```

(Item 3 normally consists of one record for each GHB. If ITMP is negative or zero, item 3 is not read. The total record length must not exceed 150 characters.)

Explanation of Parameters Read by the GHB Package

MXBND--is the maximum number of general-head boundary cells in any stress period.

IGHBCB--is a flag and a unit number.

If IGHBCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

If IGHBCB = 0, cell-by-cell flow terms will not be written.

If IGHBCB < 0, boundary leakage for each GHB cell will be written to the listing file when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.

Option--is an optional list of character values.

AUXILIARY abc or AUX abc -- "abc" is the name of an auxiliary parameter to be read for each general-head boundary as part of item 3. Up to 5 parameters can be specified, each of which must be preceded by "AUXILIARY" or "AUX." These parameters will not be used by the GHB Package, but they will be available for use by other packages. The auxiliary parameter values will be read after the Cond parameter.

CBCALLOCATE or CBC -- indicates that memory should be allocated to store cell-by-cell flow for each general-head boundary in order to make these flows available for use in other packages.

ITMP--is a flag and a counter.

If ITMP < 0, GHB data from the preceding stress period will be reused.

If ITMP ≥ 0, ITMP is the number of general-head boundaries during the current stress period.

Layer--is the layer number of the cell affected by the head-dependent boundary.

Row--is the row number of the cell affected by the head-dependent boundary.

Column--is the column number of the cell affected by the head-dependent boundary.

Bhead--is the head on the boundary.

Cond--is the hydraulic conductance of the interface between the aquifer cell and the boundary.

[xyz]--represents any auxiliary parameters for a boundary that have been defined in item 1. The auxiliary parameters must be present in each item-3 record if they are defined in item 1.

Strongly Implicit Procedure Package Input Instructions

Input to the Strongly Implicit Procedure (SIP) Package is read from the file that is type "SIP." All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, all the parameters have 10-character fields.

```
FOR EACH SIMULATION
  1. MXITER     NPARM
  2. ACCL       HCLOSE      IPCALC      WSEED       IPRSIP
```

Explanation of Parameters Read by the SIP Package

MXITER--is the maximum number of times through the iteration loop in one time step in an attempt to solve the system of finite-difference equations.

NPARM--is the number of iteration parameters to be used. Five parameters are generally sufficient.

ACCL--is the acceleration parameter. It must be greater than zero and is generally equal to one. If a zero is entered, it is changed to one.

HCLOSE--is the head change criterion for convergence. When the maximum absolute value of head change from all nodes during an iteration is less than or equal to HCLOSE, iteration stops.

IPCALC--is a flag indicating where the seed (WSEED) for calculating iteration parameters will come from.

0 - the seed entered by the user will be used.
1 - the seed will be calculated at the start of the simulation from problem parameters.

WSEED--is the seed for calculating iteration parameters. It is always read, but it is used only if IPCALC is equal to zero.

IPRSIP--is the printout interval for SIP. If IPRSIP is equal to zero, it is changed to 999. The maximum head change (positive or negative) is printed for each iteration of a time step whenever the time step is an even multiple of IPRSIP. This printout also occurs at the end of each stress period regardless of the value of IPRSIP.

Slice-Successive Overrelaxation Package Input Instructions

Input to the Slice-Successive Overrelaxation (SOR) Package is read from the file that is type "SOR." All parameters are free format if the word FREE is specified in item 4 of the Basic Package input file; otherwise, all the parameters have 10-character fields.

FOR EACH SIMULATION

```
1. MXITER
2. ACCL      HCLOSE   IPRSOR
```

Explanation of Parameters Read by the SOR Package

MXITER--is the maximum number of iterations allowed in a time step.

ACCL--is the acceleration parameter, usually between 1.0 and 2.0.

HCLOSE--is the head change criterion for convergence. When the maximum absolute value of head change from all nodes during an iteration is less than or equal to HCLOSE, iteration stops.

IPRSOR--is the printout interval for SOR. IF IPRSOR is equal to zero, it is changed to 999. The maximum head change (positive or negative) is printed for each iteration of a time step whenever the time step is an even multiple of IPRSOR. This printout also occurs at the end of each stress period regardless of the value of IPRSOR.

Input Instructions for Array Reading Utility Modules

The real two-dimensional array reader (U2DREL), the integer two-dimensional array reader (U2DINT), and the real one-dimensional array reader (U1DREL) read one array-control record and, optionally, a data array in a format specified on the array-control record. Several alternate structures for the control record are provided. The original fixed-format control records work as documented in McDonald and Harbaugh (1987), and four free-format versions have been added.

FREE-FORMAT CONTROL RECORDS FOR ARRAY READERS:

Values in bold italics are key words that can be specified as uppercase or lowercase. Each control record is limited to a length of 79 characters.

1. *CONSTANT* CNSTNT

All values in the array are set equal to CNSTNT.

2. *INTERNAL* CNSTNT FMTIN IPRN

The individual array elements will be read from the same file that contains the control record.

3. *EXTERNAL* Nunit CNSTNT FMTIN IPRN

The individual array elements will be read from the file unit number specified by Nunit. The name of the file associated with this file unit must be contained in the name file.

4. *OPEN/CLOSE* FNAME CNSTNT FMTIN IPRN

The array will be read from the file whose name is specified by FNAME. This file will be opened on unit 99 just prior to reading the array and closed immediately after the array is read. This file should not be included in the name file. A file that is read using this control record can contain only a single array. The OPEN/CLOSE option is particularly useful for running simulations that require more than 99 files using a computer that allows only 99 files to be opened simultaneously.

FIXED-FORMAT CONTROL RECORD FOR ARRAY READERS:

A fixed-format control record contains the following parameters:

```
LOCAT        CNSTNT          FMTIN        IPRN
```

LOCAT, CNSTNT, and IPRN are 10-character numeric fields. For U2DREL and U1DREL, CNSTNT is a real number. For U2DINT, CNSTNT is an integer and must not include a decimal. FMTIN is a 20-character text field. All four parameters are always read when the control record is fixed format; however, some of the parameters are unused in some situations. For example when LOCAT=0, FMTIN and IPRN are not used.

Explanation of Parameters in the Array-Control Records

CNSTNT--is a real-number constant for U2DREL and U1DREL, and an integer constant for U2DINT. If the array is being defined as a constant, CNSTNT is the constant value. If individual elements of the array are being read, the values are multiplied by CNSTNT after they are read. When CNSTNT is used as a multiplier and specified as 0, it is changed to 1.

FMTIN--is the format for reading array elements. The format must contain 20 characters or less. The format must either be a standard Fortran format that is enclosed in parentheses, "(FREE)" which indicates free format, or "(BINARY)" which indicates binary (unformatted) data. When using a free-format control record, the format must be enclosed in apostrophes if it contains one or more blanks or commas. There are only 2 ways to create a binary file that can be read by MODFLOW. The first way is to use MODFLOW to create the file by saving heads in a binary file. This is commonly done when it is desired to use computed heads from one simulation as initial heads for a subsequent simulation. The other way to create a binary file is to write a special program that generates a binary file. "(FREE)" and "(BINARY)" can only be specified in free-format control records. Also, "(BINARY)" can only be specified when using U2DREL or U2DINT, and only when the control record is EXTERNAL or OPEN/CLOSE. When the "(FREE)" option is used, be sure that all array elements have a non-blank value and that a comma or at least one blank separates adjacent values.

IPRN--is a flag that indicates if the array being read should be printed (written to the listing file) after it has been read and a code for indicating the format that should be used when it is printed. The format codes are different for each of the three array-reading modules as shown below. IPRN is set to zero when the specified value exceeds those defined. If IPRN is less than zero, the array will not be printed.

IPRN	U2DREL	U2DINT	U1DREL
0	10G11.4	10I11	10G12.5
1	11G10.3	60I1	5G12.5
2	9G13.6	40I2	
3	15F7.1	30I3	
4	15F7.2	25I4	
5	15F7.3	20I5	
6	15F7.4	10I11	
7	20F5.0	25I2	
8	20F5.1	15I4	
9	20F5.2	10I6	
10	20F5.3		
11	20F5.4		
12	10G11.4		
13	10F6.0		
14	10F6.1		
15	10F6.2		
16	10F6.3		
17	10F6.4		
18	10F6.5		

Nunit--is the unit for reading the array when the ***EXTERNAL*** free-format control record is used.

LOCAT--indicates the location of the array values for a fixed-format array control record. If LOCAT = 0, all elements are set equal to CNSTNT. If LOCAT > 0, it is the unit number for reading formatted records using FMTIN as the format. If LOCAT < 0, it is the unit number for binary (unformatted) records, and FMTIN is ignored. Also, when LOCAT is not 0, the array values are multiplied by CNSTNT after they are read.

Examples of free-format control records used to read a real array consisting of 4 rows with 7 values per row:

```
CONSTANT  5.7      This sets an entire array to the value "5.7".

INTERNAL  1.0  (7F4.0)  3     This reads the array values from the
 1.2 3.7 9.3 4.2 2.2 9.9 1.0  file that contains the control record.
 3.3 4.9 7.3 7.5 8.2 8.7 6.6  Thus, the values immediately follow the
 4.5 5.7 2.2 1.1 1.7 6.7 6.9  control record.
 7.4 3.5 7.8 8.5 7.4 6.8 8.8

EXTERNAL 52  1.0 (7F4.0)  3   This reads the array from the formatted
                              file opened on unit 52.

EXTERNAL 47  1.0 (BINARY) 3   This reads the array from the
                              binary file opened on unit 47.

OPEN/CLOSE  test.dat 1.0  (7F4.0)  3  This reads the array from the
                                      formatted file named "test.dat".
```

EXAMPLE SIMULATION

The example problem from McDonald and Harbaugh (1988) is repeated here using as many of the new features as possible. Although the simulation is hydrologically identical to the original, a number of changes were made to demonstrate features of the modified model. Output Control was added to the problem; the word method of specifying Output Control data was used. Output Control is used to save head and budget data in disk files and specify a format for printing head that creates only 80-character records. An auxiliary parameter, concentration, was included in the Well Package input data; however, this parameter is not actually used within the simulation. Input files are shown in tables 1-7. Many of the records in these files include comments to the right of the last data value in order to document the data; these comments do not interfere with the operation of MODFLOW. Such comments are allowed in the files because any characters to the right of data fields are ignored. The output listing is shown in table 8.

Table 1. -- Name File for Example Problem.

```
LIST 6 twriprob.lst
BAS 5 twriprob.bas
BCF 11 twriprob.bcf
WEL 12 twriprob.wel
DRN 13 twriprob.drn
RCH 18 twriprob.rch
SIP 19 twriprob.sip
OC 22 twriprob.oc
DATA(BINARY) 50 twriprob.bud
DATA 30 twriprob.hed
```

Table 2. -- Basic Package Input for Example Problem (twriprob.bas).

```
SAMPLE----3 LAYERS, 15 ROWS, 15 COLUMNS; STEADY STATE
CONSTANT HEADS COLUMN 1, LAYERS 1 AND 2; RECHARGE, WELLS, AND DRAINS
3 15 15 1 1  NLAY, NROW, NCOL, NPER, ITMUNI
FREE
0   -1      IAPART,ISTRT
INTERNAL  1 (15I3)  8    IBOUND LAYER 1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
INTERNAL 1 (FREE)   8    IBOUND LAYER 2
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
CONSTANT    1   IBOUND LAYER 3
999.99  HNOFLO
CONSTANT    0.  HEAD LAYER 1
CONSTANT    0.  HEAD LAYER 2
CONSTANT    0.  HEAD LAYER 3
86400.   1   1.      PERLEN,NSTP,TSMULT
```

Table 3. -- Block-Centered Flow Package Input for the Example Problem (twriprob.bcf).

```
1  50  999.99  0  0  0  0  ISS,IBCFBD,HDRY,IWDFLG,WETFCT,IWETIT,IHDWET
1 0 0  Ltype
CONSTANT   1.       TRPY
CONSTANT   5000.    DELR
CONSTANT   5000.    DELC
CONSTANT   .001     HY LAYER 1
CONSTANT   -150.    BOT LAYER 1
CONSTANT   2.E-8    Vcont LAYER 1
CONSTANT     .01    T Layer 2
CONSTANT   1.E-8    Vcont LAYER 2
CONSTANT     .02    T LAYER 3
```

Table 4. -- Well Package Input for Example Problem (twriprob.wel).

```
15  50  AUXILIARY CONCENTRATION  NWELL,IWELCB
15 ITMP
3   5  11  -5.  .2  Layer,Row,Column,Q,Concentration
2   4   6  -5.  .03
2   6  12  -5.  0.0
1   9   8  -5.  1.2
1   9  10  -5.  0.7
1   9  12  -5.  9.9
1   9  14  -5.  2.2
1  11   8  -5.  .12
1  11  10  -5.  .04
1  11  12  -5.  .4
1  11  14  -5.  .22
1  13   8  -5.  .35
1  13  10  -5.  7.6
1  13  12  -5.  8.2
1  13  14  -5.  1.1
```

Table 5. -- Drain Package Input for the Example Problem (twriprob.drn).

```
9  50  MXDRN,IDRNCB
9    ITMP
1  8   2   0.  1.E00   Layer, Row, Col, Elevation, Cond
1  8   3   0.  1.E00
1  8   4  10.  1.E00
1  8   5  20.  1.E00
1  8   6  30.  1.E00
1  8   7  50.  1.E00
1  8   8  70.  1.E00
1  8   9  90.  1.E00
1  8  10 100.  1.E00
```

Table 6. -- Recharge Package Input for the Example Problem (twriprob.rch).

```
1  50     NRCHOP,IRCHBD
1         INRECH
CONSTANT  3.E-8   RECH
```

Table 7. -- Strongly Implicit Procedure Package Input for the Example Problem (twriprob.sip).

```
50, 5   MXITER,NPARM
1., .001, 0, .001, 1  ACCL,ERR,IPCALC,WSEED,IPRSIP
```

Table 8. -- Output Control Option Input for the Example Problem (twriprob.oc).

```
HEAD PRINT FORMAT 15
HEAD SAVE FORMAT (20F10.3) LABEL
COMPACT BUDGET FILES
HEAD SAVE UNIT 30
  PERIOD 1 STEP 1
    PRINT HEAD
    PRINT BUDGET
    SAVE BUDGET
    SAVE HEAD
```

Table 9. -- Output Listing for the Example Problem (twriprob.lst).

```
 LISTING FILE: twriprob.list
                         UNIT   6

 OPENING twriprob.bas
 FILE TYPE:BAS   UNIT   5

 OPENING twriprob.bcf
 FILE TYPE:BCF   UNIT  11

 OPENING twriprob.wel
 FILE TYPE:WEL   UNIT  12

 OPENING twriprob.drn
 FILE TYPE:DRN   UNIT  13

 OPENING twriprob.rch
 FILE TYPE:RCH   UNIT  18

 OPENING twriprob.sip
 FILE TYPE:SIP   UNIT  19

 OPENING twriprob.oc
 FILE TYPE:OC   UNIT  22

 OPENING twriprob.bud
 FILE TYPE:DATA(BINARY)   UNIT  50

 OPENING twriprob.hed
 FILE TYPE:DATA(BINARY)   UNIT  30
1                                  MODFLOW
      U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER FLOW MODEL

 SAMPLE----3 LAYERS, 15 ROWS, 15 COLUMNS; STEADY STATE
 CONSTANT HEADS COLUMN 1, LAYERS 1 AND 2; RECHARGE, WELLS, AND DRAINS
 THE FREE FORMAT OPTION HAS BEEN SELECTED
    3 LAYERS        15 ROWS        15 COLUMNS
   1 STRESS PERIOD(S) IN SIMULATION
 MODEL TIME UNIT IS SECONDS

 BAS5 -- BASIC MODEL PACKAGE, VERSION 5, 1/1/95 INPUT READ FROM UNIT  5
 ARRAYS RHS AND BUFF WILL SHARE MEMORY
 INITIAL HEAD WILL BE KEPT THROUGHOUT THE SIMULATION
       6567 ELEMENTS IN X ARRAY ARE USED BY BAS
       6567 ELEMENTS OF X ARRAY USED OUT OF    4000000

 BCF5 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 5, 9/1/93 INPUT READ FROM UNIT 11
 STEADY-STATE SIMULATION
 CELL-BY-CELL FLOWS WILL BE SAVED ON UNIT 50
 HEAD AT CELLS THAT CONVERT TO DRY=   999.99
 WETTING CAPABILITY IS NOT ACTIVE
      LAYER  LAYER-TYPE CODE     INTERBLOCK T
      --------------------------------------------
         1            1          0 -- HARMONIC
         2            0          0 -- HARMONIC
         3            0          0 -- HARMONIC
        453 ELEMENTS IN X ARRAY ARE USED BY BCF
       7020 ELEMENTS OF X ARRAY USED OUT OF    4000000
 WEL5 -- WELL PACKAGE, VERSION 5, 9/1/93 INPUT READ FROM UNIT 12
 MAXIMUM OF   15 WELLS
 CELL-BY-CELL FLOWS WILL BE SAVED ON UNIT 50
 AUXILIARY WELL PARAMETER: CONCENTRATION
         75 ELEMENTS IN X ARRAY ARE USED BY WEL
       7095 ELEMENTS OF X ARRAY USED OUT OF    4000000

 DRN5 -- DRAIN PACKAGE, VERSION 5, 9/1/93 INPUT READ FROM UNIT 13
 MAXIMUM OF    9 DRAINS
 CELL-BY-CELL FLOWS WILL BE SAVED ON UNIT 50
         45 ELEMENTS IN X ARRAY ARE USED BY DRN
       7140 ELEMENTS OF X ARRAY USED OUT OF    4000000

 RCH5 -- RECHARGE PACKAGE, VERSION 5, 6/1/95 INPUT READ FROM UNIT 18
 OPTION 1 -- RECHARGE TO TOP LAYER
 CELL-BY-CELL FLOWS WILL BE SAVED ON UNIT 50
        225 ELEMENTS IN X ARRAY ARE USED BY RCH
       7365 ELEMENTS OF X ARRAY USED OUT OF    4000000

 SIP5 -- STRONGLY IMPLICIT PROCEDURE SOLUTION PACKAGE
                    VERSION 5, 9/1/93 INPUT READ FROM UNIT 19
 MAXIMUM OF  50 ITERATIONS ALLOWED FOR CLOSURE
  5 ITERATION PARAMETERS
       2905 ELEMENTS IN X ARRAY ARE USED BY SIP
      10270 ELEMENTS OF X ARRAY USED OUT OF    4000000
1
 SAMPLE----3 LAYERS, 15 ROWS, 15 COLUMNS; STEADY STATE
 CONSTANT HEADS COLUMN 1, LAYERS 1 AND 2; RECHARGE, WELLS, AND DRAINS



                     BOUNDARY ARRAY FOR LAYER   1
 READING ON UNIT   5 WITH FORMAT: (15I3)

         1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
 ...............................................................................
   1    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   2    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   3    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   4    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   5    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   6    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   7    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   8    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   9    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  10    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  11    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  12    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  13    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  14    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  15    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

```
                    BOUNDARY ARRAY FOR LAYER   2
READING ON UNIT   5 WITH FORMAT: (FREE)

        1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
..............................................................................
  1    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  2    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  3    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  4    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  5    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  6    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  7    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  8    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  9    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
 10    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
 11    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
 12    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
 13    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
 14    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
 15    -1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

          BOUNDARY ARRAY =             1 FOR LAYER   3

AQUIFER HEAD WILL BE SET TO  999.99     AT ALL NO-FLOW NODES (IBOUND=0).

            INITIAL HEAD =  0.0000000E+00 FOR LAYER   1

            INITIAL HEAD =  0.0000000E+00 FOR LAYER   2

            INITIAL HEAD =  0.0000000E+00 FOR LAYER   3

OUTPUT CONTROL IS SPECIFIED ONLY AT TIME STEPS FOR WHICH OUTPUT IS DESIRED
COMPACT CELL-BY-CELL BUDGET FILES WILL BE WRITTEN
HEAD PRINT FORMAT CODE IS  15    DRAWDOWN PRINT FORMAT CODE IS   0
HEADS WILL BE SAVED ON UNIT  30    DRAWDOWNS WILL BE SAVED ON UNIT   0

COLUMN TO ROW ANISOTROPY =   1.000000

                    DELR =   5000.000

                    DELC =   5000.000

   HYD. COND. ALONG ROWS =  0.1000000E-02 FOR LAYER   1

                  BOTTOM =  -150.0000     FOR LAYER   1

VERT HYD COND /THICKNESS =  0.2000000E-07 FOR LAYER   1

    TRANSMIS. ALONG ROWS =  0.1000000E-01 FOR LAYER   2

VERT HYD COND /THICKNESS =  0.1000000E-07 FOR LAYER   2

    TRANSMIS. ALONG ROWS =  0.2000000E-01 FOR LAYER   3
```

```
                    SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE
                    -------------------------------------------
           MAXIMUM ITERATIONS ALLOWED FOR CLOSURE =       50
                           ACCELERATION PARAMETER =     1.0000
                HEAD CHANGE CRITERION FOR CLOSURE =    0.10000E-02
                SIP HEAD CHANGE PRINTOUT INTERVAL =        1

               5 ITERATION PARAMETERS CALCULATED FROM SPECIFIED WSEED = 0.00100000 :

            0.000000E+00 0.822172E+00 0.968377E+00 0.994377E+00 0.999000E+00
          1
```

```
                              STRESS PERIOD NO.   1, LENGTH =   86400.00
                              ----------------------------------------------

                                NUMBER OF TIME STEPS =     1

                                 MULTIPLIER FOR DELT =     1.000

                              INITIAL TIME STEP SIZE =   86400.00


    15 WELLS

 LAYER   ROW   COL   STRESS RATE   WELL NO.  CONCENTRATION
 ------------------------------------------------------------
    3      5    11     -5.0000          1       0.20000
    2      4     6     -5.0000          2       0.30000E-01
    2      6    12     -5.0000          3       0.00000E+00
    1      9     8     -5.0000          4        1.2000
    1      9    10     -5.0000          5       0.70000
    1      9    12     -5.0000          6        9.9000
    1      9    14     -5.0000          7        2.2000
    1     11     8     -5.0000          8       0.12000
    1     11    10     -5.0000          9       0.40000E-01
    1     11    12     -5.0000         10       0.40000
    1     11    14     -5.0000         11       0.22000
    1     13     8     -5.0000         12       0.35000
    1     13    10     -5.0000         13        7.6000
    1     13    12     -5.0000         14        8.2000
    1     13    14     -5.0000         15        1.1000


     9 DRAINS

 LAYER   ROW   COL   ELEVATION   CONDUCTANCE   DRAIN NO.
 -------------------------------------------------------
    1      8     2   0.0000E+00     1.000           1
    1      8     3   0.0000E+00     1.000           2
    1      8     4    10.00         1.000           3
    1      8     5    20.00         1.000           4
    1      8     6    30.00         1.000           5
    1      8     7    50.00         1.000           6
    1      8     8    70.00         1.000           7
    1      8     9    90.00         1.000           8
    1      8    10    100.0         1.000           9

                 RECHARGE =  0.3000000E-07
```

```
   31 ITERATIONS FOR TIME STEP    1 IN STRESS PERIOD  1

 MAXIMUM HEAD CHANGE FOR EACH ITERATION:

    HEAD CHANGE   HEAD CHANGE   HEAD CHANGE   HEAD CHANGE   HEAD CHANGE
  LAYER,ROW,COL LAYER,ROW,COL LAYER,ROW,COL LAYER,ROW,COL LAYER,ROW,COL
 ----------------------------------------------------------------------
     -22.41         12.48         13.39         48.21         35.91
  (  3,  5, 11) (  1,  1, 15) (  3,  1, 14) (  1,  1, 15) (  3,  1, 13)
      2.482         1.430         6.214         7.411         13.66
  (  1,  9, 14) (  3, 10, 13) (  1, 12, 14) (  3, 11, 14) (  1, 15, 15)
     0.5503        0.4821        0.4711         2.019         2.302
  (  3,  8,  7) (  2,  6,  9) (  3,  5, 10) (  1, 11, 14) (  3,  5, 13)
     0.1108        0.7059E-01    0.2819        0.3141        0.3320
  (  1, 13, 12) (  3, 12, 11) (  1, 14, 14) (  3, 13, 14) (  1, 15, 15)
     0.7853E-02    0.1586E-01    0.1777E-01    0.7910E-01    0.8500E-01
  (  1, 13, 12) (  2, 11, 11) (  3, 11, 10) (  1, 14, 14) (  3,  7, 14)
     0.4169E-02    0.2555E-02    0.9769E-02    0.1082E-01    0.1030E-01
  (  1, 13, 14) (  3, 14, 15) (  1, 14, 14) (  3, 13, 14) (  1, 15, 15)
     0.2425E-03
  (  1, 13, 12)


 OUTPUT CONTROL FOR STRESS PERIOD  1   TIME STEP  1
    PRINT HEAD FOR ALL LAYERS
    PRINT BUDGET
    SAVE BUDGET
    SAVE HEAD FOR ALL LAYERS
 UBDSV2 SAVING "   CONSTANT HEAD" ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
 UBDSV1 SAVING "FLOW RIGHT FACE " ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
 UBDSV1 SAVING "FLOW FRONT FACE " ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
 UBDSV1 SAVING "FLOW LOWER FACE " ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
 UBDSV2 SAVING "           WELLS" ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
 UBDSV2 SAVING "          DRAINS" ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
 UBDSV3 SAVING "        RECHARGE" ON UNIT  50 AT TIME STEP  1, STRESS PERIOD  1
1
```

```
              HEAD IN LAYER  1 AT END OF TIME STEP  1 IN STRESS PERIOD  1
    -----------------------------------------------------------------------

         1      2      3      4      5      6      7      8      9     10
        11     12     13     14     15
 ..........................................................................
   1   0.00  24.94  44.01  59.26  71.82  82.52  91.91 100.04 106.92 112.65
     117.39 121.27 124.30 126.38 127.45
   2   0.00  24.45  43.10  57.98  70.17  80.57  90.12  98.40 105.31 110.98
     115.65 119.60 122.74 124.94 126.07
   3   0.00  23.45  41.30  55.43  66.78  76.21  86.51  95.20 102.15 107.61
     112.03 116.15 119.62 122.09 123.35
   4   0.00  21.92  38.61  51.75  61.79  68.03  81.34  90.75  97.64 102.53
     106.09 110.72 114.93 117.89 119.39
   5   0.00  19.73  34.92  47.32  57.69  66.74  77.09  85.76  92.22  96.15
      97.29 103.13 108.84 112.54 114.34
   6   0.00  16.51  29.50  40.90  51.30  61.21  71.19  79.85  86.47  90.82
      93.03  94.23 102.09 106.41 108.43
   7   0.00  11.55  21.10  31.21  41.40  51.84  63.08  72.68  79.95  84.92
      88.60  91.66  96.43  99.82 101.81
   8   0.00   3.48   6.83  16.25  26.30  36.97  52.59  64.31  72.52  77.25
      81.99  85.00  89.27  91.72  94.33
   9   0.00  10.54  19.11  28.12  36.92  45.27  52.95  55.38  65.15  66.07
      73.93  73.79  80.84  80.17  86.49
  10   0.00  14.62  25.86  35.38  43.49  50.11  54.93  57.55  62.95  65.55
      70.39  72.44  76.72  78.26  81.79
  11   0.00  17.11  29.96  40.01  47.78  53.24  55.81  53.33  60.27  59.29
      66.43  65.45  72.22  71.04  77.62
  12   0.00  18.68  32.56  43.07  50.81  55.92  58.33  58.47  61.93  63.18
      67.12  68.50  72.29  73.46  76.85
  13   0.00  19.67  34.24  45.14  53.01  58.04  59.91  56.75  62.59  60.91
      67.22  65.75  71.90  70.35  76.48
  14   0.00  20.27  35.27  46.48  54.61  60.08  63.17  64.52  67.25  68.79
      71.64  73.18  75.84  77.03  79.09
  15   0.00  20.56  35.78  47.16  55.48  61.26  65.02  67.52  69.94  72.01
      74.29  76.22  78.22  79.66  80.82
1
```

```
             HEAD IN LAYER  2 AT END OF TIME STEP  1 IN STRESS PERIOD  1
  ----------------------------------------------------------------------

         1      2      3      4      5      6      7      8      9     10
        11     12     13     14     15
.........................................................................
  1    0.00  24.66  43.73  59.02  71.61  82.32  91.72  99.86 106.75 112.49
     117.23 121.12 124.15 126.23 127.30
  2    0.00  24.17  42.83  57.74  69.95  80.36  89.93  98.22 105.14 110.81
     115.49 119.44 122.59 124.79 125.92
  3    0.00  23.17  41.03  55.19  66.53  75.77  86.29  95.02 101.98 107.43
     111.82 115.98 119.47 121.93 123.20
  4    0.00  21.65  38.34  51.50  61.35  60.17  80.90  90.55  97.45 102.27
     105.35 110.45 114.76 117.73 119.23
  5    0.00  19.48  34.65  47.07  57.44  66.30  76.85  85.57  92.00  95.41
      91.09 102.15 108.59 112.37 114.18
  6    0.00  16.27  29.24  40.65  51.07  60.98  70.98  79.65  86.28  90.54
      92.06  86.23 101.66 106.23 108.27
  7    0.00  11.38  20.95  31.05  41.25  51.70  62.90  72.48  79.76  84.73
      88.35  91.24  96.22  99.65 101.64
  8    0.00   4.21   8.33  17.58  27.58  38.25  52.94  64.19  72.34  77.12
      81.81  84.86  89.10  91.59  94.17
  9    0.00  10.38  18.96  27.98  36.79  45.16  52.86  56.13  65.08  66.79
      73.87  74.48  80.77  80.84  86.38
 10    0.00  14.40  25.61  35.15  43.27  49.91  54.76  57.48  62.79  65.49
      70.24  72.37  76.57  78.20  81.64
 11    0.00  16.87  29.70  39.78  47.56  53.05  55.68  54.09  60.20  60.04
      66.37  66.18  72.16  71.75  77.51
 12    0.00  18.43  32.31  42.85  50.60  55.73  58.16  58.41  61.78  63.12
      66.98  68.44  72.15  73.40  76.69
 13    0.00  19.42  33.98  44.91  52.80  57.85  59.78  57.50  62.53  61.65
      67.16  66.48  71.84  71.06  76.37
 14    0.00  20.02  35.02  46.26  54.41  59.88  62.99  64.39  67.08  68.66
      71.48  73.06  75.68  76.91  78.93
 15    0.00  20.30  35.52  46.94  55.28  61.07  64.84  67.34  69.76  71.84
      74.11  76.04  78.04  79.49  80.65
1
```

```
                  HEAD IN LAYER  3 AT END OF TIME STEP  1 IN STRESS PERIOD  1
  ---------------------------------------------------------------------------

              1       2       3       4       5       6       7       8       9      10
             11      12      13      14      15
 ............................................................................
   1      1.80   24.34   43.36   58.70   71.33   82.06   91.48   99.63  106.53  112.27
        117.02  120.91  123.95  126.04  127.10
   2      1.76   23.85   42.46   57.42   69.66   80.07   89.68   97.99  104.91  110.59
        115.27  119.23  122.39  124.59  125.72
   3      1.69   22.86   40.67   54.87   66.20   75.28   85.98   94.77  101.74  107.19
        111.52  115.74  119.25  121.73  123.00
   4      1.58   21.35   37.98   51.17   60.85   62.69   80.41   90.28   97.19  101.88
        104.11  110.04  114.52  117.52  119.02
   5      1.42   19.18   34.30   46.75   57.10   65.80   76.54   85.30   91.67   94.17
         77.46  100.67  108.24  112.14  113.96
   6      1.18   15.99   28.91   40.33   50.76   60.67   70.70   79.38   86.01   90.12
         90.60   88.55  101.18  105.98  108.05
   7      0.83   11.21   20.79   30.88   41.09   51.55   62.67   72.22   79.50   84.46
         87.98   90.77   95.94   99.41  101.41
   8      0.43    5.13   10.19   19.27   29.19   39.84   53.40   64.07   72.11   76.95
         81.58   84.68   88.88   91.44   93.95
   9      0.75   10.22   18.82   27.84   36.66   45.06   52.78   57.03   65.02   67.64
         73.81   75.31   80.72   81.64   86.24
  10      1.04   14.13   25.29   34.85   42.99   49.65   54.54   57.44   62.61   65.44
         70.05   72.33   76.39   78.15   81.43
  11      1.22   16.59   29.37   39.47   47.28   52.79   55.53   55.01   60.16   60.94
         66.33   67.06   72.13   72.60   77.38
  12      1.34   18.15   31.97   42.54   50.32   55.47   57.94   58.37   61.60   63.08
         66.80   68.41   71.97   73.36   76.49
  13      1.42   19.14   33.65   44.61   52.53   57.60   59.63   58.39   62.48   62.54
         67.12   67.35   71.80   71.90   76.24
  14      1.46   19.73   34.68   45.96   54.13   59.63   62.76   64.24   66.87   68.52
         71.27   72.91   75.47   76.77   78.71
  15      1.48   20.01   35.18   46.63   55.00   60.81   64.59   67.11   69.52   71.61
         73.87   75.82   77.81   79.27   80.42

 HEAD WILL BE SAVED ON UNIT  30 AT END OF TIME STEP  1, STRESS PERIOD  1
1
```

```
 VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP  1 IN STRESS PERIOD  1
 ------------------------------------------------------------------------------

    CUMULATIVE VOLUMES      L**3       RATES FOR THIS TIME STEP      L**3/T
    ------------------                 ------------------------

          IN:                                      IN:
          ---                                      ---
      CONSTANT HEAD =          0.0000        CONSTANT HEAD =            0.0000
              WELLS =          0.0000                WELLS =            0.0000
             DRAINS =          0.0000               DRAINS =            0.0000
           RECHARGE =   13607999.0000             RECHARGE =          157.5000

           TOTAL IN =   13607999.0000             TOTAL IN =          157.5000

         OUT:                                     OUT:
         ----                                     ----
      CONSTANT HEAD =    4326521.5000        CONSTANT HEAD =           50.0755
              WELLS =    6480000.0000                WELLS =           75.0000
             DRAINS =    2801080.5000               DRAINS =           32.4199
           RECHARGE =          0.0000             RECHARGE =            0.0000

          TOTAL OUT =   13607602.0000            TOTAL OUT =          157.4954

           IN - OUT =        397.0000             IN - OUT =        4.5929E-03

PERCENT DISCREPANCY =          0.00     PERCENT DISCREPANCY =          0.00




          TIME SUMMARY AT END OF TIME STEP  1 IN STRESS PERIOD  1
                    SECONDS      MINUTES       HOURS        DAYS         YEARS
                    -----------------------------------------------------------
   TIME STEP LENGTH  86400.       1440.0       24.000      1.0000     2.73785E-03
 STRESS PERIOD TIME  86400.       1440.0       24.000      1.0000     2.73785E-03
         TOTAL TIME  86400.       1440.0       24.000      1.0000     2.73785E-03
```

REFERENCES

American National Standards Institute, 1966, FORTRAN: American National Standards for Information Processing, X3.9-1966, 36 p.

American National Standards Institute, 1978, Programming language FORTRAN: American National Standards for Information Processing, X3.9-1978, 18 ch.

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. and McDonald, M.G., 1996, Programmer's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 96-486, 220 p.

McDonald, M.G., and Harbaugh, A.W., 1984, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-FIle Report 83-875, 528 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, chap. 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.