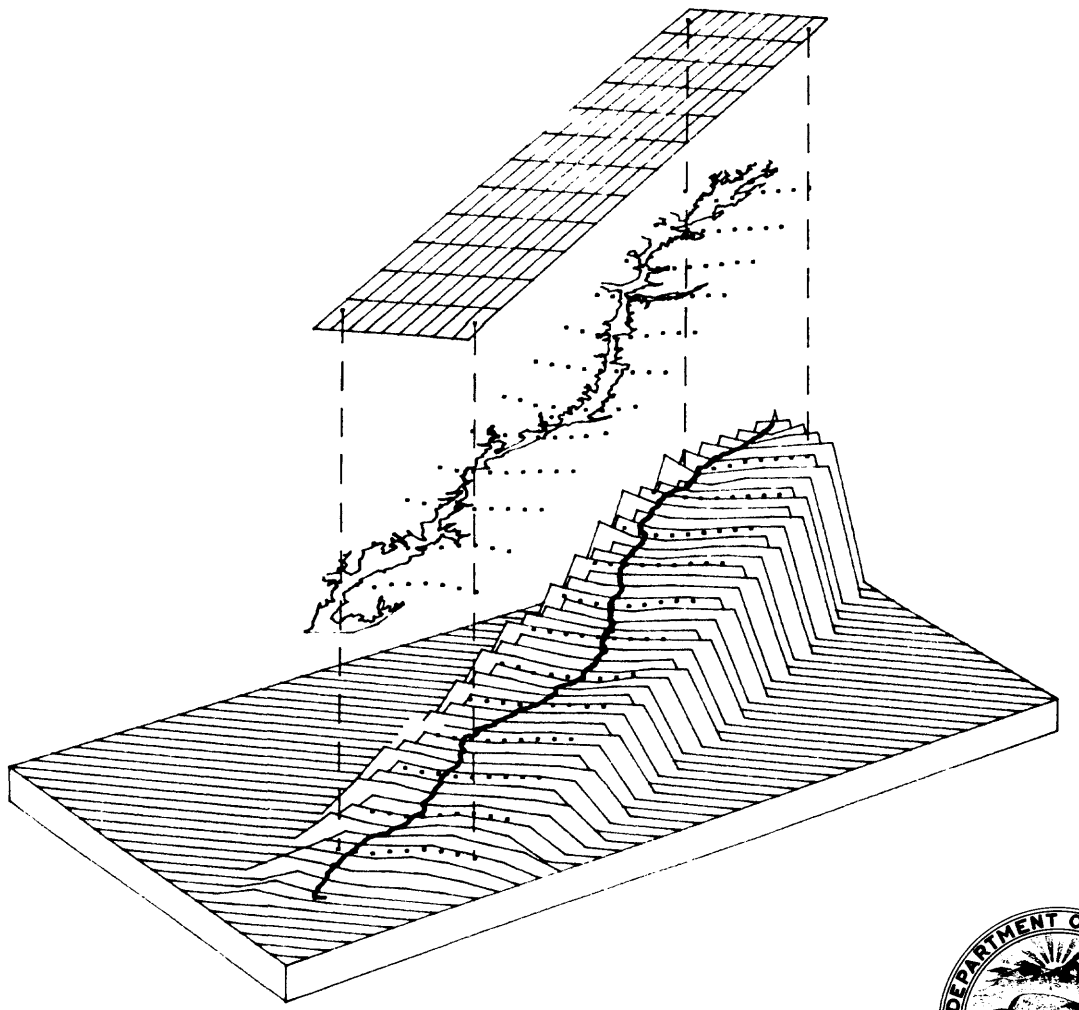
STRUCTURE AND APPLICATION OF AN INTERFACE PROGRAM BETWEEN A GEOGRAPHIC-INFORMATION SYSTEM AND A GROUND-WATER FLOW MODEL

U.S. GEOLOGICAL SURVEY

Open-File Report 90-165



STRUCTURE AND APPLICATION OF AN INTERFACE PROGRAM BETWEEN A GEOGRAPHIC-INFORMATION SYSTEM AND A GROUND-WATER FLOW MODEL

By Peter C. Van Metre

U.S. GEOLOGICAL SURVEY

Open-File Report 90-165

Tucson, Arizona
May 1990

Cover: Overlay of a ground-water model grid with digital coverages to develop model data sets using geographic-information system techniques. Upper layer is a polygon coverage of the model grid. Middle layer is a line coverage of the extent of an aquifer and a point coverage of center points of the model grid. Lower layer is a surface coverage of land-surface elevation, a line coverage of a stream, and a point coverage of center points of the model grid.

UNITED STATES DEPARTMENT OF THE INTERIOR

MANUEL LUJAN, JR., *Secretary*

GEOLOGICAL SURVEY

Dallas L. Peck, *Director*

Copies of the computer program and test data sets on tape or diskette are available at cost of processing from:

U.S. Geological Survey
National Water Information System
437 National Center
Reston, VA 22092
Telephone: (703) 648-5695

For additional information write to:

District Chief
U.S. Geological Survey
Federal Building, FB-44
300 West Congress Street
Tucson, Arizona 85701-1393

Copies of this report can be purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Federal Center, Building 810
Box 25425
Denver, Colorado 80225

CONTENTS

ILLUSTRATIONS

STRUCTURE AND APPLICATION OF AN INTERFACE PROGRAM BETWEEN A GEOGRAPHIC-INFORMATION SYSTEM AND A GROUND-WATER FLOW MODEL

By

Peter C. Van Metre

ABSTRACT

A computer-program interface between a geographic-information system and a ground-water flow model links two unrelated software systems for use in developing the flow models. The interface program allows the modeler to compile and manage geographic components of a ground-water model within the geographic-information system. A significant savings of time and effort is realized in developing, calibrating, and displaying the ground-water flow model. Four major guidelines were followed in developing the interface program: (1) no changes to the ground-water flow model code were to be made, (2) a data structure was to be designed within the geographic-information system that follows the same basic data structure as the ground-water flow model, (3) the interface program was to be flexible enough to support all options available within the model, and (4) the interface program was to be as efficient as possible in terms of computer time used and online-storage space needed. Because some programs in the interface are written in control-program language, the interface will run only on a computer with the PRIMOS operating system.

INTRODUCTION

A computer-program interface between a geographic-information system (GIS) and a ground-water flow model links two unrelated software systems for use in developing the flow models. The geographic-information system, ARC/INFO[1], is a digital-mapping and data-management system that is used to develop and manage geographic data bases. ARC/INFO was interfaced in this study with the modular three-dimensional finite-difference ground-water flow model of McDonald and Harbaugh (1984 and 1988). The interface program is designed to enter, store, update, and output all controlling and geographic information from ARC/INFO that is necessary to run the ground-water flow model. The interface program is written for a PRIME computer. Use of the interface on a different system such as a VAX would entail modifying programs written in control-program language (CPL) to ARC macro language (AML) or some other supported macro language.

---

[1]Use of the brand or trade names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

In ground-water modeling, the initial data collection, compilation, interpretation, and integration are time-consuming tasks. For example, in order to develop a data file that defines the top of an aquifer, plotted well locations and their associated geologic information are hand contoured and overlaid on a plot of the model grid. Values are interpolated and assigned to the grid, and the grid values are then input to a computer data file called an array. The geographic and data-base management capabilities of ARC/INFO can be used to automate most of these tasks.

ARC/INFO also can be used to edit arrays during calibration, input various stresses for predictive model runs, and create report products. Geographic data in ARC/INFO are edited interactively on a computer-graphics terminal. The user selects model nodes or areas of nodes by pointing to them or defining a box around them with the cursor on the screen. Parameter values of selected nodes then can be changed with a calculate command. Changes can also be made using INFO commands that are independent of ARC. INFO data files associated with an ARC/INFO coverage can be reselected for subsets of those files, and variables can be recalculated. A program in the ARC/INFO system called ARCPLOT offers a convenient way of plotting both input data and model results.

The interface program performs no data transformations or manipulations and is used only to conveniently store and move data between ARC/INFO and the ground-water flow model. A working knowledge of the ground-water flow model and ARC/INFO is necessary to understand the relation between those programs and the interface program. Documentation of the interface program includes a listing and description of the INFO data files and a listing and description of all INFO, control-program language (CPL), and Fortran programs. A glossary of terms is included at the end of this report for the reader who may be unfamiliar with the terms used in the ground-water model and ARC/INFO.

The purpose of this report is to document an interface program that can be used to move data between a GIS and a ground-water flow model. Four major guidelines were followed in developing the interface program: (1) no changes to the ground-water flow model code were to be made, (2) a structure was to be designed within the GIS that follows the same basic structure as the ground-water flow model, (3) the interface program was to be flexible enough to support all options available within the model, and (4) the interface program was to be as efficient as possible in terms of computer time used and online-storage space needed. Application of the interface program was tested using a two-column by nine-row by four-layer ground-water flow model (fig. 1). The author wishes to thank Dean Anderson of Environmental Systems Research Institute for his valuable advice in devising file structure and INFO programs in this study.

STRUCTURE OF THE INTERFACE

The interface program was developed on the PRIME computer (Seybold, 1985) and uses a tree-structure system for directories within the program (fig. 2). Directories of the interface program include the following:

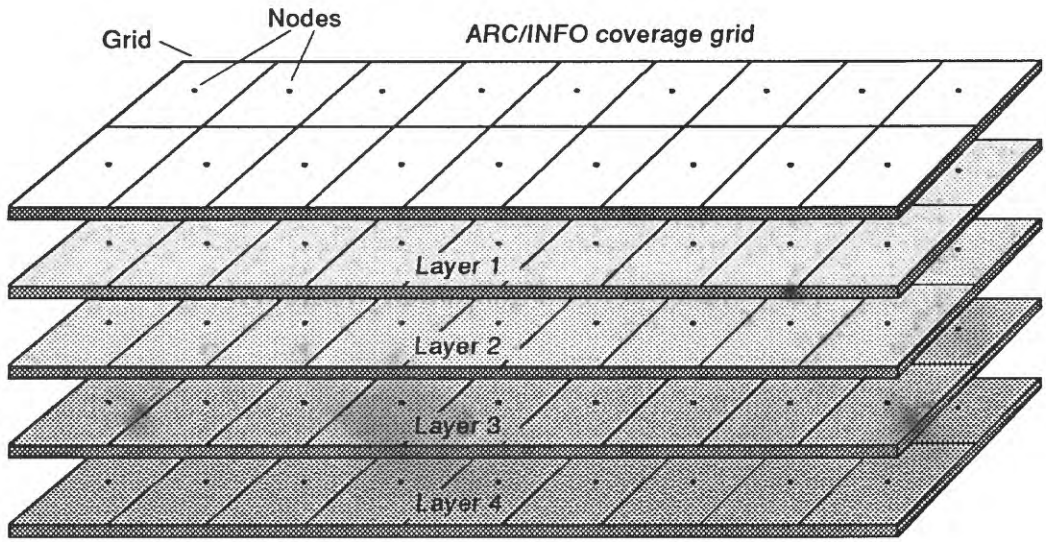


Figure 1.—ARC/INFO coverages, subroutines GRID and NODES.

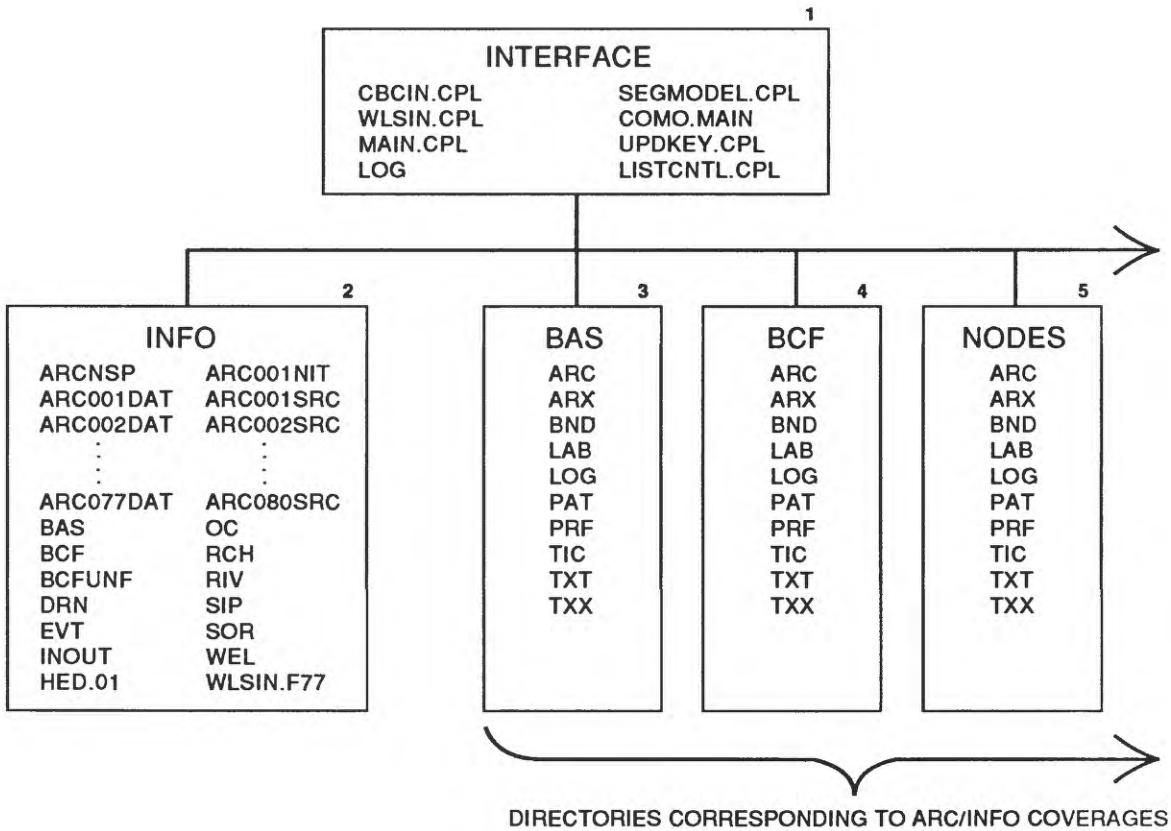


Figure 2.—Organization of PRIMOS directories for ARC/INFO and the interface program.

- Directory 1, which is named Interface, holds the command programming languages (CPLs) that drive and execute the interface program. The output of model data from ARC/INFO, the running of the model, and the return of model-output values to ARC/INFO can be executed from this directory.

- Directory 2 is created by ARC/INFO to hold INFO files associated with all coverages (Environmental Systems Research Institute, 1987). Data files associated with coverages and INFO programs that are part of the interface are accessed within INFO at this level. Those files can be viewed by attaching to INFO, entering the command "INFO", and specifying INFO user name "ARC." INFO files in this directory can also be viewed by attaching to directory 1, entering the command "ARC", and then entering the command "INFO".

  Directory 2 can have two distinct sets of files, those within INFO and those at the PRIMOS level. By default, the PRIMOS data files associated with each model package are in directory 2, and the model output is written in directory 2. The user can view ASCII model-output files before generating ARC/INFO coverages of model output in this directory.

- Directories 3-5 are PRIMOS directories that are created by ARC/INFO to hold data files associated with coverages. Each of these directories holds files for one coverage and has the same name as that coverage.

Flow of the Interface Program

The interface program is controlled by MAIN.CPL, WLSIN.CPL, and CBCIN.CPL. The three CPLs reside and are executed from directory 1 (fig. 1) and are described in detail in the sections "Control of Package Output Programs" and "Returning the Model Output to ARC/INFO." A brief narrative of the flow of a model run using the interface program is as follows.

To initiate a model run, the user invokes MAIN.CPL, which performs the following steps:

1. Enters INFO and determines which packages are in use and which of those packages will require new PRIMOS data files.

2. Runs the INFO package output program for each of those packages creating a new PRIMOS data file for each package at the PRIMOS level in directory 2 (fig. 1).

3. Runs SEGMODEL.CPL to open all necessary data files and runs the model.

4. Runs Fortran program WLSIN.F77, which opens the output control (OC) PRIMOS file and selected other PRIMOS data files, determines what was saved (unformatted) on disk, then reads and reformats those records for input to INFO.

Program execution then is stopped to allow the user to view model output, such as the statistics generated for a model run, before generating coverages of model-output heads, drawdowns, or flow terms. To generate coverage files of those outputs, the user runs either WLSIN.CPL for heads and drawdowns or CBCIN.CPL for flow terms. The argument required for each CPL is the name of the reformatted model-output file generated by WLSIN.F77.

Steps performed by WLSIN.CPL and CBCIN.CPL:

1. Determine if a coverage already exists by the argument name.

2. If so, bring the specified data into INFO and use it to overwrite that attribute in the coverage .PAT file. If a coverage does not exist, use ARC commands to create a new coverage by that name by copying coverage NODES, then bring the specified data into INFO and add it to the new coverage .PAT file.

Description of Package Output Programs and INFO Data Files

McDonald and Harbaugh (1988) includes 13 packages. Two of the packages are undefined; therefore, they are not included in the interface program. The 11 packages, their abbreviations, and their IUNIT locations are given in the table on p. 6.

Discussion of the packages in this report is based on similar structure of the packages and not on a numerical order. Each package has a PRIMOS data-file structure as defined by McDonald and Harbaugh (1984 and 1988). Each PRIMOS data file can contain control information, one-dimensional arrays, two-dimensional arrays, and lists of data by layer, row, and column. Control information can include number of layers, rows, and columns or maximum number of wells for a particular stress period. One-dimensional arrays may be arrays such as the TRPY array in the block-centered flow package that are constant through a simulation. Two-dimensional arrays can be arrays such as the IBOUND array in the basic package; some two-dimensional arrays can be constant through a simulation or can vary for different stress periods.

The basic (BAS), block-centered flow (BCF), recharge (RCH), and evapotranspiration (EVT) packages can contain two-dimensional arrays that are number of rows times number of columns in length. An item or variable output from INFO will, by default, appear as a list with one value per line. Because the arrays output for each of those four packages are held as items in INFO data files, the most efficient way to output them to PRIMOS data files is as lists. Lists, however, are not convenient to work with in the PRIMOS file structure, particularly if the model is large. An option was added to the interface program to write arrays across the page.

The BCF package can contain one-dimensional arrays—either number of layers in length, number of rows in length, or number of columns in length. By default, those arrays are written from INFO across the page

Package	Abbreviation	IUNIT location	Default IUNIT value
Basic	BAS	--	5
Block-centered flow	BCF	1	29
Well	WEL	2	30
Drain	DRN	3	31
River	RIV	4	32
Evapotranspiration	EVT	5	33
Undefined package	---	6	--
General-head boundary	GHB	7	35
Recharge	RCH	8	36
Strongly implicit procedure	SIP	9	37
Undefined package	---	10	--
Slice-successive overrelaxation	SOR	11	--
Output control	OC	12	38

with a page width of 132 columns. The well (WEL), drain (DRN), river (RIV), and general-head boundary (GHB) packages can contain lists of data by layer, row, and column and are stored and output from INFO as lists. The strongly implicit procedure (SIP), slice-successive overrelaxation (SOR), and output-control (OC) packages contain only controlling information. Data that are not arrays are defined in the INFO data structure as single 80-character-wide items with redefined items for each model variable. Output from INFO is executed as whole files with either the "save compress" command or the "display print" command.

Basic Package

Three INFO data files—control (BAS.CNTL), formats (BAS.FMTS), and attribute (BAS.PAT)—are used to hold the data required for the BAS package. The control information contained in the first five cards in the BAS package (McDonald and Harbaugh, 1988) is contained in the BAS.CNTL data file. The array-control card for each two-dimensional (row by column) array is held in the BAS.FMTS file. Data arrays that are not represented by a constant value in the corresponding array-control card are in the BAS.PAT file. The BAS.PAT file is the point-attribute file for a point coverage in ARC/INFO that contains the center point of every block in the model grid, and, therefore, is number of rows times number of columns in length. One additional file, BAS.KEY, is used to generate output for the BAS package. BAS.KEY includes KEY, BAS#, IKEY, and ROW and contains one record for each model node. KEY is a unique integer that corresponds to the record number for the BAS.PAT file when it is sorted on row and column. BAS# is the unique integer associated with each node in the BAS.PAT file; however, it generally will not match KEY.

INFO File Structure and Example Data Files

```
DATAFILE NAME: BAS.CNTL                                          12/5/1988
   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT                80   80  C    -
     **  REDEFINED ITEMS  **
   1  CD1.HEADNG          80   80  C    -
   1  CD2.HEADNG          48   48  C    -
   1  NLAY                10   10  I    -
  11  NROW                10   10  I    -
  21  NCOL                10   10  I    -
  31  NPER                10   10  I    -
  41  ITMUNI              10   10  I    -
   1  IUNIT1               3    3  I    -
   4  IUNIT2               3    3  I    -
   7  IUNIT3               3    3  I    -
  10  IUNIT4               3    3  I    -
  13  IUNIT5               3    3  I    -
  16  IUNIT6               3    3  I    -
  19  IUNIT7               3    3  I    -
  22  IUNIT8               3    3  I    -
  25  IUNIT9               3    3  I    -
  28  IUNIT10              3    3  I    -
  31  IUNIT11              3    3  I    -
  34  IUNIT12              3    3  I    -


   1  IAPART              10   10  I    -
  11  ISTRT               10   10  I    -


ENTER COMMAND >LI
$RECNO   TEXT
     1   TEST RUN OF ARC/INFO-GWMDL WITH CONTROL CARDS IN INFO
     2   **4/1/87**
     3             4         2         9         1         1
     4    29 30 31 32 33    35 36 37       38
     5            0         1

DATAFILE NAME: BAS.FMTS                                          12/5/1988

   6 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  LOCAT               10   10  I    -
  11  CNSTNT              10   10  C    -
  21  FMTIN               20   20  C    -
  41  IPRN                10   10  I    -
  51  ARRAY               10   10  C    -
  61  IKEY                 2    2  I    -
     **  REDEFINED ITEMS  **
   2  CARD                59   59  C    -
  11  NSTP                10   10  I    -
  21  TSMULT              10   10  N    2
```

```
ENTER COMMAND >LI
$RECNO        LOCAT CNSTNT      FMTIN                    IPRN ARRAY    IKEY
     1            5              (20I4)                   -1 IBOUND1      9
     2            5              (20I4)                   -1 IBOUND2      9
     3            5              (20I4)                   -1 IBOUND3      9
     4            5              (20I4)                   -1 IBOUND4      9
     5          999                                       0 HNOFLO        9
     6            5              (20F6.0)                 -1 SHEAD1       9
     7            5              (20F6.0)                 -1 SHEAD2       9
     8            5              (20F6.0)                 -1 SHEAD3       9
     9            5              (20F6.0)                 -1 SHEAD4       9
    10        86400          1       1.00                  0              0

DATAFILE NAME: BAS.PAT                                              12/5/1988

  14 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
  1  AREA                  4   12  F     3
  5  PERIMETER             4   12  F     3
  9  BAS#                  4    5  B     -
 13  BAS-ID                4    5  B     -
 17  KEY                   5    5  I     -
 22  IBOUND1               3    3  I     -
 25  IBOUND2               3    3  I     -
 28  IBOUND3               3    3  I     -
 31  IBOUND4               3    3  I     -
 34  SHEAD1                5    5  N     0
 39  SHEAD2                5    5  N     0
 44  SHEAD3                5    5  N     0
 49  SHEAD4                5    5  N     0
 54  IKEY                  2    2  I     -
    **  REDEFINED ITEMS  **
 13  GRID#                 4    5  B     -
```

```
ENTER COMMAND >LI
$RECNO   AREA  PERIMETER BAS# BAS-ID KEY IBOUND1 IBOUND2 IBOUND3 IBOUND4 SHEAD1 SHEAD2 SHEAD3 SHEAD4 IKEY
     1  0.000     0.000    1      1   1       1       1       1       1    254    263    268    271    9
     2  0.000     0.000    2      2   2       1       1       1       1    267    270    273    274    9
     3  0.000     0.000    3      3   3       1       1       1       1    274    278    280    281    9
     4  0.000     0.000    4      4   4       0       1       1       1      0    288    288    289    9
     5  0.000     0.000    5      7   7       0       0       1       1      0      0    308    311    9
     6  0.000     0.000    6      8   8       0       0       0       1      0      0      0    318    9
     7  0.000     0.000    7      9   9       0       0       0       1      0      0      0    320    9
     8  0.000     0.000    8     10  10       1       1       1       1    254    263    268    271    9
     9  0.000     0.000    9     11  11       1       1       1       1    267    270    273    274    9
    10  0.000     0.000   10     12  12       1       1       1       1    274    278    280    281    9
    11  0.000     0.000   11     13  13       0       1       1       1      0    288    288    289    9
    12  0.000     0.000   12     14  14       0       1       1       1      0    294    296    296    9
    13  0.000     0.000   13     15  15       0       0       1       1      0      0    304    304    9
    14  0.000     0.000   14     16  16       0       0       1       1      0      0    308    311    9
    15  0.000     0.000   15     17  17       0       0       0       1      0      0      0    318    9
    16  0.000     0.000   16     18  18       0       0       0       1      0      0      0    320    9
    17  0.000     0.000   17      5   5       0       1       1       1      0    294    296    296    9
    18  0.000     0.000   18      6   6       0       0       1       1      0      0    304    304    9
```

```
DATAFILE NAME: BAS.KEY                                          12/5/1988

   4 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  KEY                 5    5  I    -
   6  BAS#                4    5  B    -
  10  IKEY                2    2  I    -
  12  ROW                 5    5  I    -

ENTER COMMAND >LI
$RECNO     KEY  BAS# IKEY   ROW
     1       1     1  9       1
     2       2     2  9       1
     3       3     3  9       1
     4       4     4  9       1
     5       5    17  9       1
     6       6    18  9       1
     7       7     5  9       1
     8       8     6  9       1
     9       9     7  9       1
    10      10     8  9       2
    11      11     9  9       2
    12      12    10  9       2
    13      13    11  9       2
    14      14    12  9       2
    15      15    13  9       2
    16      16    14  9       2
    17      17    15  9       2
    18      18    16  9       2
```

Narrative of Basic Output Program

The basic package output program (BASOUT.PG) generates a PRIMOS file named BAS for a model run. The generated BAS file can be used for model input and requires no modification to the ground-water flow model. The program is executed in the following manner:

1. Select the BAS.CNTL file and use "SAVE BAS COMPRESS INIT" to open a PRIMOS file named BAS, initialize that file, and write all selected records in BAS.CNTL to that file.

2. Select BAS.FMTS and relate to files BAS.KEY and BAS.PAT if any arrays are nonconstant.

3. Loop through each record in the BAS.FMTS file to test for FMTIN that contain parentheses. Each record in BAS.FMTS is either an array-control card, HNOFLO card, or a PERLEN, NSTP, and TSMULT data card (McDonald and Harbaugh, 1988). In other package-output programs where the .FMTS file contains array-control cards only, the test at this step is

for LOCAT greater than zero. If FMTIN does not contain parentheses, the array is represented by a constant value and only the array-control card is output. If FMTIN contains parentheses, a format is being given to read an array. Then the value of LOCAT is tested. If LOCAT equals 5, the default FORTRAN unit number for PRIMOS data file BAS, the array is output to PRIMOS file BAS. If LOCAT does not equal 5, a PRIMOS file named with the value in INFO item ARRAY is opened and the array is output to that file. In this way, the array control cards can be loaded to PRIMOS file BAS and the arrays can be loaded to separate PRIMOS files. Either LIST or WIDE output format is specified when executing the interface program. The effect of each is as follows:

a. If LIST output format is specified, the array-control card is output and the corresponding array in BAS.PAT is output as a list.

b. If WIDE output format is specified and each row of data will fit within 132 characters, the array-control card is output and the corresponding array in BAS.PAT is output across the page.

c. If WIDE output format is specified and each row of data will not fit within 132 characters, the array-control card is output, BAS.KEY is selected and related to BAS.PAT, and rows are reselected one at a time and output across the page wrapping to the subsequent lines as necessary. BAS.FMTS is then selected and the relate sequence in step 2 is re-established. The test model in this situation took two to three times more central-processing unit (CPU) and input and output (I/O) time to generate files as a list format or the wide format described in b. above.

ARRAY and IKEY are two items in the .FMTS files that are not found in the model documentation. ARRAY is the character name of the array that the array-control card corresponds to and must match the item name of that array in the corresponding .PAT file. The value stored in ARRAY is used to output its corresponding data values from the .PAT file. IKEY is an integer item that equals 9 in all files and is used as part of the file-relating sequence in INFO that is used for all packages that produce two-dimensional arrays.

In the test model, SORT was added to the BCF.FMTS data file and is an integer equal to the record number of the .FMTS file when it is properly sorted. If new array-control cards are added or if the order needs to be changed for any reason, values of SORT can be updated and the file resorted.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: BASOUT.PG
10003 REM *This program is run from either OUTALL or OUTSEL
10004 REM *to output necessary data for the MacDonald-
10005 REM *Harbaugh modular model.
10006 REM
10007  FORMAT $NUM13,8,I
10008  FORMAT $NUM14,8,I
10009  FORMAT $NUM15,8,I
10010  FORMAT $NUM30,3,I
10011  FORMAT $NUM31,3,I
10012  FORMAT $NUM32,3,I
10013  FORMAT $NUM35,3,I
10014  FORMAT $NUM36,3,I
10015  FORMAT $NUM37,3,I
10016  FORMAT $NUM38,3,I
10017  FORMAT $CHR16,40,C
10018  FORMAT $CHR21,40,C
10019  FORMAT $CHR39,4,C
10020 SEL BAS.CNTL
10021 SAVE BAS COMPRESS INIT
10022 OUTPUT BAS
10023 SEL BAS.FMTS
10024 CA $NUM13 = 1
10025 CA $NUM14 = $NOSEL
10026 RES FMTIN CN '('
10027  IF $NOSEL GT 0
10028   REL BAS.KEY 1 BY IKEY SEQ
10029   REL BAS.PAT 2 BY $1BAS# 0
10030  ENDIF
10031 ASEL
10032 DO UNTIL $NUM13 GT $NUM14
10033 CONCAT $CHR16 FROM 'RES BY $RECNO = ',$NUM13
10034 EXEC $CHR16
10035 IF FMTIN CN '('
10036  DIS CARD PR
10037  IF LOCAT NC '5'
10038   CONCAT $CHR16 FROM 'OUTPUT ',ARRAY
10039   EXEC $CHR16
10040  ENDIF
10041  IF $CHR39 EQ 'LIST'
10042   CONCAT $CHR21 FROM 'DIS $2',ARRAY,' PR'
10043   NEXT
10044   EXEC $CHR21
10045   NEXT OFF
10046  ELSE
10047   REM *Wide format, determine width and write arrays
10048   RUN FMTIN.PG LINK
10049  IF $PRINTER-SIZE GT 132
10050   CALC $PRINTER-SIZE = 132
10051   CONCAT $CHR21 FROM 'DIS $1',ARRAY,',= PR'
10052   SEL BAS.KEY
```

```
 10053   REL BAS.PAT BY BAS# 0
 10054   CALC $NUM35 = 1
 10055   DO UNTIL $NUM35 GT $NUM36
 10056   CONCAT $CHR16 FROM 'RES BY ROW = ',$NUM35
 10057   EXEC $CHR16
 10058   EXEC $CHR21
 10059   CALC $NUM15 = $NUM35
 10060   CALC $NUM35 = $NUM15 + 1
 10061   ASEL
 10062   DOEND
 10063   SEL BAS.FMTS
 10064  ELSE
 10065   NEXT
 10066   CONCAT $CHR21 FROM 'DIS $2',ARRAY,',= PR'
 10067   EXEC $CHR21
 10068   NEXT OFF
 10069  ENDIF
 10070  ENDIF
 10071 OUTPUT BAS
 10072 ELSE
 10073  DIS CARD PR
 10074 ENDIF
 10075 ASEL
 10076  CALC $NUM15 = $NUM13
 10077  CALC $NUM13 = $NUM15 + 1
 10078 DOEND
```

Listing of Program FMTIN.PG

```
 10000  PROGRAM SECTION ONE
 10001 FORMAT $NUM37,3,I
 10002 IF FMTIN CN 'F'
 10003  IF FMTIN CN 'F4'
 10004   CALC $PRINTER-SIZE = 4 * $NUM37
 10005  ENDIF
 10006  IF FMTIN CN 'F5'
 10007   CALC $PRINTER-SIZE = 5 * $NUM37
 10008  ENDIF
 10009  IF FMTIN CN 'F6'
 10010   CALC $PRINTER-SIZE = 6 * $NUM37
 10011  ENDIF
 10012  IF FMTIN CN 'F7'
 10013   CALC $PRINTER-SIZE = 7 * $NUM37
 10014  ENDIF
 10015  IF FMTIN CN 'F8'
 10016   CALC $PRINTER-SIZE = 8 * $NUM37
 10017  ENDIF
 10018  IF FMTIN CN 'F9'
 10019   CALC $PRINTER-SIZE = 9 * $NUM37
 10020  ENDIF
 10021  IF FMTIN CN 'F10'
 10022   CALC $PRINTER-SIZE = 10 * $NUM37
 10023  ENDIF
 10024  IF FMTIN CN 'F11'
```

```
10025   CALC $PRINTER-SIZE = 11 * $NUM37
10026  ENDIF
10027  IF FMTIN CN 'F12'
10028   CALC $PRINTER-SIZE = 12 * $NUM37
10029  ENDIF
10030  IF FMTIN CN 'F13'
10031   CALC $PRINTER-SIZE = 13 * $NUM37
10032  ENDIF
10033 ELSE
10034  IF FMTIN CN 'I2'
10035   CALC $PRINTER-SIZE = 2 * $NUM37
10036  ENDIF
10037  IF FMTIN CN 'I3'
10038   CALC $PRINTER-SIZE = 3 * $NUM37
10039  ENDIF
10040  IF FMTIN CN 'I4'
10041   CALC $PRINTER-SIZE = 4 * $NUM37
10042  ENDIF
10043  IF FMTIN CN 'I5'
10044   CALC $PRINTER-SIZE = 5 * $NUM37
10045  ENDIF
10046  IF FMTIN CN 'I6'
10047   CALC $PRINTER-SIZE = 6 * $NUM37
10048  ENDIF
10049  IF FMTIN CN 'I7'
10050   CALC $PRINTER-SIZE = 7 * $NUM37
10051  ENDIF
10052 ENDIF
```

Block-Centered Flow Package

A storage and output structure similar to BAS is used for the BCF package. Three additional INFO files for one-dimensional arrays are included in BCF. Only one-dimensional arrays that are not constant need to have the corresponding data file defined in INFO.

INFO File Structure and Example Data Files

```
DATAFILE NAME: BCF.CNTL                                          12/5/1988
   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT               80   80  C    -
     **  REDEFINED ITEMS  **
   1  ISS                10   10  I    -
  11  IBCFCB             10   10  I    -
   1  LAYCON1             2    2  I    -
   3  LAYCON2             2    2  I    -
   5  LAYCON3             2    2  I    -
   7  LAYCON4             2    2  I    -

ENTER COMMAND >LI
$RECNO   TEXT
     1            1        41
     2    1 3 3 3
```

DATAFILE NAME: BCF.FMTS 12/5/1988

7 ITEMS: STARTING IN POSITION 1

COL	ITEM NAME	WDTH	OPUT	TYP	N.DEC	ALTERNATE NAME
1	LOCAT	10	10	I	-	
11	CNSTNT	10	10	C	-	
21	FMTIN	20	20	C	-	
41	IPRN	10	10	I	-	
51	ARRAY	10	10	C	-	
61	IKEY	2	2	I	-	
63	SORT#	2	2	I	-	
	** REDEFINED ITEMS **					
2	CARD	59	59	C	-	

ENTER COMMAND >LI

$RECNO	LOCAT	CNSTNT	FMTIN	IPRN	ARRAY	IKEY	SORT#
1	0	1.		0	TRPY	9	1
2	0	.528E4		0	DELR	9	2
3	0	.528E4		-1	DELC	9	3
4	0	.10E-2		-1	TRAN1	9	4
5	29	1.	(20F6.0)	-1	BOT1	9	5
6	0	.10E-8		-1	VCONT1	9	6
7	0	.70E-3		-1	TRAN2	9	7
8	29	1.	(20F6.0)	-1	BOT2	9	8
9	0	.90E-9		-1	VCONT2	9	9
10	29	1.	(20F6.0)	-1	TOP2	9	10
11	0	.60E-3		-1	TRAN3	9	11
12	29	1.	(20F6.0)	-1	BOT3	9	12
13	0	.80E-9		-1	VCONT3	9	13
14	29	1.	(20F6.0)	-1	TOP3	9	14
15	0	.50E-3		-1	TRAN4	9	15
16	29	1.	(20F6.0)	-1	BOT4	9	16
17	29	1.	(20F6.0)	-1	TOP4	9	17

DATAFILE NAME: BCF.PAT 12/5/1988

13 ITEMS: STARTING IN POSITION 1

COL	ITEM NAME	WDTH	OPUT	TYP	N.DEC	ALTERNATE NAME
1	AREA	4	12	F	3	
5	PERIMETER	4	12	F	3	
9	BCF#	4	5	B	-	
13	BCF-ID	4	5	B	-	
17	KEY	5	5	I	-	
22	BOT1	5	5	N	0	
27	BOT2	5	5	N	0	
32	TOP2	5	5	N	0	
37	BOT3	5	5	N	0	
42	TOP3	5	5	N	0	
47	BOT4	5	5	N	0	
52	TOP4	5	5	N	0	
57	IKEY	2	2	I	-	
	** REDEFINED ITEMS **					
13	GRID#	4	5	B	-	

```
ENTER COMMAND >LI
$RECNO        AREA      PERIMETER  BCF#  BCF-ID  KEY  BOT1  BOT2  TOP2  BOT3  TOP3  BOT4  TOP4  IKEY
     1  199,934.800          0.000     1       1    1   230   206   230   178   206   146   178     0
     2  199,934.800          0.000     2       2    2   240   218   240   191   218   160   191     9
     3  199,934.800          0.000     3       3    3   250   230   250   204   230   174   204     9
     4  199,934.800          0.000     4       4    4     0   242   275   217   242   188   217     9
     5  199,934.800          0.000     5       5    5     0   254   280   230   254   202   230     9
     6  199,934.800          0.000     6       6    6     0     0     0   243   285   216   243     9
     7  199,934.800          0.000     7       7    7     0     0     0   256   290   230   256     9
     8  199,934.800          0.000     8       8    8     0     0     0     0     0   244   295     9
     9  199,934.800          0.000     9       9    9     0     0     0     0     0   258   300     9
    10  199,934.800          0.000    10      10   10   230   206   230   178   206   146   178     9
    11  199,934.800          0.000    11      11   11   240   218   240   191   218   160   191     9
    12  199,934.800          0.000    12      12   12   250   230   250   204   230   174   204     9
    13  199,934.800          0.000    13      13   13     0   242   275   217   242   188   217     9
    14  199,934.800          0.000    14      14   14     0   254   280   230   254   202   230     9
    15  199,934.800          0.000    15      15   15     0     0     0   243   285   216   243     9
    16  199,934.800          0.000    16      16   16     0     0     0   256   290   230   256     9
    17  199,934.800          0.000    17      17   17     0     0     0     0     0   244   295     9
    18  199,934.800          0.000    18      18   18     0     0     0     0     0   258   300     9

DATAFILE NAME: BCF.KEY                                              12/5/1988

   4 ITEMS: STARTING IN POSITION     1
COL   ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  KEY                  5    5  I     -
   6  BCF#                 4    5  B     -
  10  IKEY                 2    2  I     -
  12  ROW                  3    3  I     -

ENTER COMMAND >LI
$RECNO      KEY  BCF# IKEY ROW
     1        1     1  9     1
     2        2     2  9     1
     3        3     3  9     1
     4        4     4  9     1
     5        5     5  9     1
     6        6     6  9     1
     7        7     7  9     1
     8        8     8  9     1
     9        9     9  9     1
    10       10    10  9     2
    11       11    11  9     2
    12       12    12  9     2
    13       13    13  9     2
    14       14    14  9     2
    15       15    15  9     2
    16       16    16  9     2
    17       17    17  9     2
    18       18    18  9     2
```

Narrative of Block-Centered Flow Output Program

1. Select the BCF.CNTL file and use "SAVE BCF COMPRESS INIT" to open PRIMOS file named BCF, initialize that file, and write all selected records in BCF.CNTL to that file.

2. Select BCF.FMTS and loop through the first three records that are the array-control cards for the one-dimensional arrays TRPY, DELR, and DELC. If FMTIN contains parentheses, the array is not constant and values are expected. Relate data file TRPY, DELR, or DELC, depending on the value of item ARRAY, and output the array-control card and its associated data array. If FMTIN does not contain parentheses, output array-control card only.

3. Establish a file relate between the selected file BCF.FMTS and the files BCF.KEY and BCF.PAT if any arrays are nonconstant.

4. Loop through each remaining record in BCF.FMTS testing for FMTIN that contain parentheses. If FMTIN contains parentheses, the array is not constant and values are expected. The value of LOCAT is then tested. If LOCAT equals 29, the default FORTRAN unit number for BCF, the array is output to PRIMOS file BCF. If LOCAT does not equal 29, a PRIMOS file named with the value in INFO item ARRAY is opened, for example BOT1 for the array describing the bottom of layer one, and the array is output to that file. Either LIST or WIDE output format is specified when executing the interface program. The effect of each is as follows:

   a. If LIST output format is specified, the array-control card is output and the corresponding array in BCF.PAT is output as a list.

   b. If WIDE output format is specified and each row of data will fit within 132 characters, the array-control card is output and the corresponding array in BCF.PAT is output across the page.

   c. If WIDE output format is specified and each row of data will not fit within 132 characters, the array-control card is output, BCF.KEY is selected and related to BCF.PAT, and rows are reselected one at a time and output across the page, wrapping to subsequent lines as necessary. BAS.FMTS is then selected and the relate sequence in step 2 is re-established.

   If FMTIN does not contain parentheses, the array is represented by a constant value and only the array-control card is output.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: BCFOUT.PG
10003 REM *This program is run from OUTALL if IUNIT(1)
10004 REM *is greater than zero or from OUTSEL if
10005 REM *response was Y(es).
10006 REM
10007  FORMAT $NUM13,8,I
10008  FORMAT $NUM14,8,I
10009  FORMAT $NUM15,8,I
10010  FORMAT $NUM30,3,I
10011  FORMAT $NUM31,3,I
10012  FORMAT $NUM32,3,I
10013  FORMAT $NUM35,3,I
10014  FORMAT $NUM36,3,I
10015  FORMAT $NUM37,3,I
10016  FORMAT $NUM38,3,I
10017  FORMAT $CHR16,40,C
10018  FORMAT $CHR21,40,C
10019  FORMAT $CHR39,4,C
10020 CALC $PRINTER-SIZE = 132
10021 SEL BCF.CNTL
10022 SAVE BCF COMPRESS INIT
10023 OUTPUT BCF
10024 SEL BCF.FMTS
10025 REM *Output TRPY, DELR, and DELC.
10026 CA $NUM13 = 1
10027 CA $NUM14 = 3
10028 DO UNTIL $NUM13 GT $NUM14
10029 RES BY $RECNO = $NUM13
10030  IF FMTIN CN '('
10031   DIS CARD PR
10032   CONCAT $CHR16 FROM 'REL ',ARRAY,' BY IKEY SEQ'
10033   EXEC $CHR16
10034   CONCAT $CHR21 FROM 'DIS $1',ARRAY,',= PR'
10035     NEXT
10036   EXEC $CHR21
10037     NEXT OFF
10038  ELSE
10039   DIS CARD PR
10040  ENDIF
10041 ASEL
10042  CA $NUM15 = $NUM13
10043  CA $NUM13 = $NUM15 + 1
10044 DOEND
10045 CA $NUM13 = 4
10046 CA $NUM14 = $NOSEL
10047 RES $RECNO GT 3 AND LOCAT GT 0
10048  IF $NOSEL GT 0
10049   REL BCF.KEY 1 BY IKEY SEQ
10050   REL BCF.PAT 2 BY $1BCF# 0
10051  ENDIF
10052 ASEL
```

```
10053 DO UNTIL $NUM13 GT $NUM14
10054 CONCAT $CHR16 FROM 'RES BY $RECNO = ',$NUM13
10055 EXEC $CHR16
10056 IF LOCAT GT 0
10057  DIS CARD PR
10058  IF LOCAT NC '29'
10059   CONCAT $CHR16 FROM 'OUTPUT ',ARRAY
10060   EXEC $CHR16
10061  ENDIF
10062  IF $CHR39 EQ 'LIST'
10063   CONCAT $CHR21 FROM 'DIS $2',ARRAY,' PR'
10064   NEXT
10065   EXEC $CHR21
10066   NEXT OFF
10067  ELSE
10068   REM *Wide format, determine width and write arrays.
10069   RUN FMTIN.PG LINK
10070  IF $PRINTER-SIZE GT 132
10071   CALC $PRINTER-SIZE = 132
10072   CONCAT $CHR21 FROM 'DIS $1',ARRAY,',= PR'
10073   SEL BCF.KEY
10074   REL BCF.PAT BY BCF# 0
10075   CALC $NUM35 = 1
10076   DO UNTIL $NUM35 GT $NUM36
10077   CONCAT $CHR16 FROM 'RES BY ROW = ',$NUM35
10078   EXEC $CHR16
10079   EXEC $CHR21
10080   CALC $NUM15 = $NUM35
10081   CALC $NUM35 = $NUM15 + 1
10082   ASEL
10083   DOEND
10084   SEL BCF.FMTS
10085  ELSE
10086   NEXT
10087   CONCAT $CHR21 FROM 'DIS $2',ARRAY,',= PR'
10088   EXEC $CHR21
10089   NEXT OFF
10090  ENDIF
10091  ENDIF
10092 OUTPUT BCF
10093 ELSE
10094  DIS CARD PR
10095 ENDIF
10096 ASEL
10097  CALC $NUM15 = $NUM13
10098  CALC $NUM13 = $NUM15 + 1
10099 DOEND
```

Recharge Package

The recharge (RCH) package and the evapotranspiration (EVT) packages are similar in that they both can contain a different set of data arrays for each stress period simulated. This capability differs from the BAS and BCF packages, which require only one set of arrays for each simulation. In addition, both RCH and EVT can have a variable number of arrays for any given stress period—0, 1, or 2 for RCH and 0, 1, 2, 3, or 4 for EVT.

Each unique set of nonconstant arrays to be used for one or more stress periods is held as a separate INFO data file instead of an ARC/INFO point coverage. This reduces the amount of on-line disk space required to hold the data. Each data file is number of rows times number of columns in length and is sorted on row and column. The file that contains values for a particular modeled stress period is specified in item KPER in the RCH.CNTL file. If KPER contains the value RCH.PER3, data file RCH.PER3 will be related to RCH.CNTL and array values will be output.

Each unique data array for RCH or EVT can be output for as many different stress periods as desired. Each file has the structure of a key file built into it for faster relating and to control the order of data output.

INFO File Structure and Example Data Files

```
DATAFILE NAME: RCH.CNTL                                          12/9/1988

   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT               62   62  C    -
     **  REDEFINED ITEMS  **
   1  NRCHOP             10   10  I    -
  11  IRCHCB             10   10  I    -
   1  INRECH             10   10  I    -
  11  INIRCH             10   10  I    -
  21  KPER               10   10  C    -
   1  LOCAT              10   10  I    -
  11  CNSTNT             10   10  C    -
  21  FMTIN              20   20  C    -
  41  IPRN               10   10  I    -
  51  ARRAY              10   10  C    -
  61  IKEY                2    2  I    -
   2  CARD               61   61  C    -


ENTER COMMAND >LI
$RECNO   TEXT
     1            2        47
     2            1         1RCH.PER1
     3              .26E-8                                  -1RECH
     4           36         1 (20I4)                        -1IRCH       9
```

DATAFILE NAME: RCH.PER1 12/9/1988

6 ITEMS: STARTING IN POSITION 1

COL	ITEM NAME	WDTH	OPUT	TYP	N.DEC	ALTERNATE NAME
1	KEY	5	5	I	-	
6	RECH	6	6	N	2	
12	IRCH	3	3	I	-	
15	IKEY	2	2	I	-	
17	RCH#	4	5	B	-	
21	ROW	3	3	I	-	

ENTER COMMAND >LI

$RECNO	KEY	RECH	IRCH	IKEY	RCH#	ROW
1	1	0.00	1	9	1	1
2	2	0.00	1	9	2	1
3	3	0.00	1	9	3	1
4	4	0.00	2	9	4	1
5	5	0.00	2	9	17	1
6	6	0.00	3	9	18	1
7	7	0.00	3	9	5	1
8	8	0.00	4	9	6	1
9	9	0.00	4	9	7	1
10	10	0.00	1	9	8	2
11	11	0.00	1	9	9	2
12	12	0.00	1	9	10	2
13	13	0.00	2	9	11	2
14	14	0.00	2	9	12	2
15	15	0.00	3	9	13	2
16	16	0.00	3	9	14	2
17	17	0.00	4	9	15	2
18	18	0.00	4	9	16	2

Note: Item KEY in INFO file RCH.PER1 is not needed by the output program but can be used to relate this file to a coverage of the model grid (NODES or GRID). KEY is equal to record number when the model grid data file (NODES.PAT or GRID.PAT) is sorted on row, column.

Narrative of Recharge Output Program

1. Select the RCH.CNTL file, open PRIMOS file RCH with INIT, reselect for first record and output to PRIMOS file RCH.

2. Reselect for the second record in RCH.CNTL, output that record to RCH, and determine if arrays will be output (INRECH and (or) INIRCH greater than or equal to 0). There will be one card with INRECH and INIRCH for each stress period modeled. If INRECH or INIRCH are greater than or equal to zero, loop through control card for each array (the next one or two records in RCH.CNTL).

3. If LOCAT is greater than zero, the array is not constant and values are expected. Relate data file specified in item KPER of RCH.CNTL. Either LIST or WIDE output format is specified when executing the interface program. The effect of each is as follows:

   a. If LIST output format is specified, the array-control card is output and the corresponding array in the related data file is output as a list.

   b. If WIDE output format is specified and each row of data will fit within 132 characters, the array-control card is output and the corresponding array in the related data file is output across the page.

   c. If WIDE output format is specified and each row of data will not fit within 132 characters, the array-control card is output, the data file specified in KPER is selected, and rows are reselected one at a time and output across the page, wrapping to subsequent lines as necessary.

   If LOCAT is less than or equal to zero, only the array-control card is output.

4. Select RCH.CNTL, if necessary, and reselect for control card for next stress period. Repeat steps two and three.

5. Repeat steps 2, 3, and 4 as necessary.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: RCHOUT.PG
10003 REM *This program is run from OUTALL if IUNIT(8)
10004 REM *is greater than zero or from OUTSEL if
10005 REM *response was Y(es).
10006 REM
10007  FORMAT $NUM13,8,I
10008  FORMAT $NUM14,8,I
10009  FORMAT $NUM15,8,I
10010  FORMAT $NUM30,3,I
10011  FORMAT $NUM31,3,I
10012  FORMAT $NUM32,3,I
10013  FORMAT $NUM35,3,I
10014  FORMAT $NUM36,3,I
10015  FORMAT $NUM37,3,I
10016  FORMAT $NUM38,3,I
10017  FORMAT $CHR16,40,C
10018  FORMAT $CHR21,40,C
10019  FORMAT $CHR26,16,C
```

```
10020  FORMAT $CHR39,4,C
10021 CALC $PRINTER-SIZE = 132
10022 SEL RCH.CNTL
10023 OUTPUT RCH INIT
10024 CA $NUM13 = 2
10025 CA $NUM14 = $NOSEL
10026 RES BY $RECNO EQ 1
10027 DIS CARD PRINT
10028 CA $NUM32 = NRCHOP
10029 ASEL
10030 DO UNTIL $NUM13 GT $NUM14
10031  REM *Reselect for cd2 and set flags
10032 RES BY $RECNO EQ $NUM13
10033 CA $NUM30 = INRECH
10034 CA $NUM31 = INIRCH
10035 MOVE KPER TO $CHR26
10036 DIS CARD PRINT
10037 CONCAT $CHR16 FROM 'REL ',$CHR26,' IKEY SEQ'
10038  IF $NUM30 GE 0
10039 REM *read new rech array
10040  CALC $NUM15 = $NUM13
10041  CALC $NUM13 = $NUM15 + 1
10042  ASEL
10043  RES BY $RECNO EQ $NUM13
10044 IF LOCAT GT 0
10045  EXEC $CHR16
10046  DIS CARD PR
10047  IF $CHR39 EQ 'LIST'
10048   CONCAT $CHR21 FROM 'DIS $1',ARRAY,' PR'
10049   NEXT
10050   EXEC $CHR21
10051   NEXT OFF
10052  ELSE
10053   REM *Wide format, determine width and write arrays.
10054   RUN FMTIN.PG LINK
10055  IF $PRINTER-SIZE GT 132
10056   CALC $PRINTER-SIZE = 132
10057   CONCAT $CHR16 FROM 'SEL ',$CHR26
10058   CONCAT $CHR21 FROM 'DIS ',ARRAY,',= PR'
10059   EXEC $CHR16
10060   CALC $NUM35 = 1
10061   DO UNTIL $NUM35 GT $NUM36
10062   CONCAT $CHR16 FROM 'RES BY ROW = ',$NUM35
10063   EXEC $CHR16
10064   EXEC $CHR21
10065   CALC $NUM15 = $NUM35
10066   CALC $NUM35 = $NUM15 + 1
10067   ASEL
10068   DOEND
10069   SEL RCH.CNTL
10070  ELSE
10071   NEXT
10072   CONCAT $CHR21 FROM 'DIS $1',ARRAY,',= PR'
10073   EXEC $CHR21
10074   NEXT OFF
```

```
10075   CALC $PRINTER-SIZE = 132
10076  ENDIF
10077  ENDIF
10078 ELSE
10079  DIS CARD PR
10080 ENDIF
10081 ENDIF
10082 IF $NUM32 = 2
10083 REM *IRCH array expected
10084  IF $NUM31 GE 0
10085 REM *read new irch array
10086  CALC $NUM15 = $NUM13
10087  CALC $NUM13 = $NUM15 + 1
10088  ASEL
10089  RES $RECNO EQ $NUM13
10090 IF LOCAT GT 0
10091  EXEC $CHR16
10092  DIS CARD PR
10093  IF $CHR39 EQ 'LIST'
10094   CONCAT $CHR21 FROM 'DIS $1',ARRAY,' PR'
10095   NEXT
10096   EXEC $CHR21
10097   NEXT OFF
10098  ELSE
10099   REM *Wide format, determine width and write arrays.
10100  RUN FMTIN.PG LINK
10101  IF $PRINTER-SIZE GT 132
10102   CALC $PRINTER-SIZE = 132
10103   CONCAT $CHR16 FROM 'SEL ',$CHR26
10104   CONCAT $CHR21 FROM 'DIS ',ARRAY,',= PR'
10105   EXEC $CHR16
10106   CALC $NUM35 = 1
10107   DO UNTIL $NUM35 GT $NUM36
10108   CONCAT $CHR16 FROM 'RES BY ROW = ',$NUM35
10109   EXEC $CHR16
10110   EXEC $CHR21
10111   CALC $NUM15 = $NUM35
10112   CALC $NUM35 = $NUM15 + 1
10113   ASEL
10114   DOEND
10115   SEL RCH.CNTL
10116  ELSE
10117   NEXT
10118   CONCAT $CHR21 FROM 'DIS $1',ARRAY,',= PR'
10119   EXEC $CHR21
10120   NEXT OFF
10121   CALC $PRINTER-SIZE = 132
10122  ENDIF
10123  ENDIF
10124 ELSE
10125  DIS CARD PR
10126 ENDIF
10127 ENDIF
10128 ENDIF
10129  CALC $NUM15 = $NUM13
```

```
10130  CALC $NUM13 = $NUM15 + 1
10131  ASEL
10132 DOEND
```

Evapotranspiration Package

A file (EVT.CNTL) that contains package control information and array-control cards is used in conjunction with a variable number of data files. Each unique set of nonconstant arrays, to be used for one or more stress periods modeled, is held as an INFO data file. The file containing values that are desired for a particular modeled stress period is specified in item KPER in the EVT.CNTL file. If KPER contains the value, EVT.PER3, data file EVT.PER3 will be related to EVT.CNTL, and array values will be output.

INFO File Structure and Example Data Files

```
DATAFILE NAME: EVT.CNTL                                          12/ 5/1988


   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT               62   62  C    -
     **  REDEFINED ITEMS  **
   1  NEVTOP             10   10  I    -
  11  IEVTCB             10   10  I    -
   1  INSURF             10   10  I    -
  11  INEVTR             10   10  I    -
  21  INEXDP             10   10  I    -
  31  INIEVT             10   10  I    -
  41  KPER               10   10  C    -
   1  LOCAT              10   10  I    -
  11  CNSTNT             10   10  C    -
  21  FMTIN              20   20  C    -
  41  IPRN               10   10  I    -
  51  ARRAY              10   10  C    -
  61  IKEY                2    2  I    -
   2  CARD               61   61  C    -


ENTER COMMAND >LI


$RECNO   TEXT
     1            2        45
     2            1         1         1        1EVT.PER1
     3           33         1 (20F7.0)                    -1SURF       9
     4             .91E-8                                 -1EVTR
     5            10.                                     -1EXDP
     6           33         1 (20I4)                      -1IEVT       9
```

DATAFILE NAME: EVT.PER1 12/5/1988

8 ITEMS: STARTING IN POSITION 1

COL	ITEM NAME	WDTH	OPUT	TYP	N.DEC	ALTERNATE NAME
1	KEY	5	5	I	-	
6	SURF	6	6	N	1	
12	EVTR	6	6	N	2	
18	EXDP	6	6	N	1	
24	IEVT	3	3	I	-	
27	IKEY	2	2	I	-	
29	EVT#	4	5	B	-	
33	ROW	3	3	I	-	

ENTER COMMAND >LI

$RECNO	KEY	SURF	EVTR	EXDP	IEVT	IKEY	EVT#	ROW
1	1	260.0	0.00	0.0	1	9	1	1
2	2	265.0	0.00	0.0	1	9	2	1
3	3	270.0	0.00	0.0	1	9	3	1
4	4	275.0	0.00	0.0	2	9	4	1
5	5	280.0	0.00	0.0	2	9	17	1
6	6	285.0	0.00	0.0	3	9	18	1
7	7	290.0	0.00	0.0	3	9	5	1
8	8	295.0	0.00	0.0	4	9	6	1
9	9	300.0	0.00	0.0	4	9	7	1
10	10	260.0	0.00	0.0	1	9	8	2
11	11	265.0	0.00	0.0	1	9	9	2
12	12	270.0	0.00	0.0	1	9	10	2
13	13	275.0	0.00	0.0	2	9	11	2
14	14	280.0	0.00	0.0	2	9	12	2
15	15	285.0	0.00	0.0	3	9	13	2
16	16	290.0	0.00	0.0	3	9	14	2
17	17	295.0	0.00	0.0	4	9	15	2
18	18	300.0	0.00	0.0	4	9	16	2

Note: Item KEY in INFO file EVT.PER1 is not used by the output program but can be used to relate this file to a coverage of the model grid (NODES or GRID). KEY is equal to record number when the model grid data file (NODES.PAT or GRID.PAT) is sorted on row, column.

Narrative of Evapotranspiration Output Program

1. Select the EVT.CNTL file, open PRIMOS file EVT with INIT, reselect for the first record and output it to PRIMOS file EVT.

2. Reselect for the second record in EVT.CNTL, output that record to EVT, and determine if arrays will be output (INSURF, INEVTR, INEXDP, or INIEVT read flags greater than or equal to 0). There will be one card with either the first three or all four read flags for each stress period

modeled depending on the variable NEVTOP. If the read flags indicate a control card and possibly an array will be read by the model, then loop through each of the subsequent control cards (the next one, two, three of four records in EVT.CNTL).

3. If LOCAT is greater than zero, the array is not constant and values are expected. Relate EVT.CNTL to the data file specified in item KPER. Either LIST or WIDE output format is specified when executing the interface program. The effect of each is as follows:

   a. If LIST output format is specified, the array-control card is output and the corresponding array in the related data file is output as a list.

   b. If WIDE output format is specified and each row of data will fit within 132 characters, the array-control card is output and the corresponding array in the related data file is output across the page.

   c. If WIDE output format is specified and each row of data will not fit within 132 characters, the array-control card is output, the data file specified in KPER is selected, and rows are reselected one at a time and output across the page wrapping to subsequent lines as necessary.

   If LOCAT is less than or equal to zero, only the array-control card is output.

4. Select EVT.CNTL, if necessary, and reselect for control card for next stress period. Repeat steps two and three.

5. Repeat steps two, three, and four as necessary.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: EVTOUT.PG
10003 REM *This program is run from OUTALL if IUNIT(5)
10004 REM *is greater than zero or from OUTSEL if
10005 REM *response was Y(es).
10006 REM
10007  FORMAT $NUM13,8,I
10008  FORMAT $NUM14,8,I
10009  FORMAT $NUM15,8,I
```

```
10010  FORMAT $NUM30,3,I
10011  FORMAT $NUM31,3,I
10012  FORMAT $NUM32,3,I
10013  FORMAT $NUM33,3,I
10014  FORMAT $NUM34,3,I
10015  FORMAT $NUM35,3,I
10016  FORMAT $NUM36,3,I
10017  FORMAT $NUM37,3,I
10018  FORMAT $NUM38,3,I
10019  FORMAT $CHR16,40,C
10020  FORMAT $CHR21,40,C
10021  FORMAT $CHR26,16,C
10022  FORMAT $CHR39,4,C
10023 CALC $PRINTER-SIZE = 132
10024 SEL EVT.CNTL
10025 OUTPUT EVT INIT
10026 CA $NUM13 = 2
10027 CA $NUM14 = $NOSEL
10028 RES BY $RECNO EQ 1
10029 DIS CARD PRINT
10030 CA $NUM34 = NEVTOP
10031 ASEL
10032 DO UNTIL $NUM13 GT $NUM14
10033  REM *Reselect for card 2 and set flags
10034 RES BY $RECNO = $NUM13
10035 CA $NUM30 = INSURF
10036 CA $NUM31 = INEVTR
10037 CA $NUM32 = INEXDP
10038 CA $NUM33 = INIEVT
10039 MOVE KPER TO $CHR26
10040 DIS CARD PRINT
10041 CONCAT $CHR16 FROM 'REL ',KPER,' IKEY SEQ'
10042  IF $NUM30 GE 0
10043   REM *read new SURF array
10044   RUN EVT.SUB LINK
10045  ENDIF
10046  IF $NUM31 GE 0
10047   REM *read new EVTR array
10048   RUN EVT.SUB LINK
10049  ENDIF
10050  IF $NUM32 GE 0
10051   REM *read new EXDP array
10052   RUN EVT.SUB LINK
10053  ENDIF
10054  IF $NUM34 = 2
10055   REM *option two, IEVT array expected
10056   IF $NUM33 GE 0
10057    REM *read new IEVT array
10058    RUN EVT.SUB LINK
10059   ENDIF
10060  ENDIF
10061  CALC $NUM15 = $NUM13
10062  CALC $NUM13 = $NUM15 + 1
10063  ASEL
10064 DOEND
```

Listing of a Subroutine Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: EVT.SUB
10003 REM *This program is a subroutine called from EVTOUT>P.PG to
10004 REM *output control cards and arrays.
10005 REM
10006  FORMAT $NUM13,8,I
10007  FORMAT $NUM14,8,I
10008  FORMAT $NUM15,8,I
10009  FORMAT $NUM30,3,I
10010  FORMAT $NUM31,3,I
10011  FORMAT $NUM32,3,I
10012  FORMAT $NUM33,3,I
10013  FORMAT $NUM34,3,I
10014  FORMAT $NUM35,3,I
10015  FORMAT $NUM36,3,I
10016  FORMAT $NUM37,3,I
10017  FORMAT $NUM38,3,I
10018  FORMAT $CHR16,40,C
10019  FORMAT $CHR21,40,C
10020  FORMAT $CHR26,16,C
10021  FORMAT $CHR39,4,C
10022  CALC $NUM15 = $NUM13
10023  CALC $NUM13 = $NUM15 + 1
10024  ASEL
10025  RES BY $RECNO EQ $NUM13
10026 IF LOCAT GT 0
10027  EXEC $CHR16
10028  DIS CARD PR
10029  IF $CHR39 EQ 'LIST'
10030   CONCAT $CHR21 FROM 'DIS $1',ARRAY,' PR'
10031   NEXT
10032   EXEC $CHR21
10033   NEXT OFF
10034  ELSE
10035   REM *Wide format, determine width and write arrays.
10036   RUN FMTIN.PG LINK
10037  IF $PRINTER-SIZE GT 132
10038   CALC $PRINTER-SIZE = 132
10039   CONCAT $CHR16 FROM 'SEL ',$CHR26
10040   CONCAT $CHR21 FROM 'DIS ',ARRAY,',= PR'
10041 DIS $CHR26
10042 DIS $CHR16
10043   EXEC $CHR16
10044   CALC $NUM35 = 1
10045   DO UNTIL $NUM35 GT $NUM36
10046   CONCAT $CHR16 FROM 'RES BY ROW = ',$NUM35
10047   EXEC $CHR16
10048   EXEC $CHR21
10049   CALC $NUM15 = $NUM35
10050   CALC $NUM35 = $NUM15 + 1
10051   ASEL
10052   DOEND
```

```
10053   SEL EVT.CNTL
10054  ELSE
10055   CONCAT $CHR21 FROM 'DIS $1',ARRAY,',= PR'
10056   NEXT
10057   EXEC $CHR21
10058   NEXT OFF
10059   CALC $PRINTER-SIZE = 132
10060  ENDIF
10061  ENDIF
10062 ELSE
10063  DIS CARD PR
10064 ENDIF
```

In addition to the output program, an interactive update program was written for the EVT package. It uses INFO program language and INFO input forms to allow the user to easily add or update records for the file EVT.CNTL. To invoke the update sequence, issue the command 'RUN EVT.UPD' from within the INFO workspace.

Listing of Update Program

```
10000  PROGRAM SECTION ONE
10001 SEL EVT.CNTL
10002 LIST
10003 FORMAT $NUM1,4,I
10004 FORMAT $CHR2,1,C
10005 FORMAT $NUM3,4,I
10006 FORMAT $NUM4,4,I
10007 LABEL ENTER
10008  DIS 'DO YOU WISH TO ADD A RECORD OR UPDATE A RECORD? (A/U) '
10009  ACCEPT $CHR2
10010 DIS $CHR2
10011 IF $CHR2 NE 'A' AND $CHR2 NE 'U'
10012  GOTO END
10013 ENDIF
10014 IF $CHR2 EQ 'A'
10015  GOTO ADD
10016 ENDIF
10017 LABEL UPDAGN
10018 DIS 'ENTER $RECNO YOU WISH TO UPDATE (QUIT=99) ',=
10019 ACCEPT $NUM1
10020 IF $NUM1 EQ 99
10021  GOTO END
10022 ENDIF
10023 IF $NUM1 LT 1 OR $NUM1 GT $NOSEL
10024  DIS 'RECORD ',$NUM1,' DOES NOT EXIST'
10025  GOTO ENTER
10026 ENDIF
10027 RES $RECNO = $NUM1
10028 IF $NUM1 = 1
10029  UPD W EVT.CD1.IPF LN
10030 REM *MUST HAVE ARRAY NAMES LOADED TO WORK.
10031 ELSE
```

```
 10032  IF ARRAY GT ' '
 10033   UPD W FMTS.IPF LN
 10034  ELSE
 10035   UPD W EVT.CD2.IPF LN
 10036  ENDIF
 10037 ENDIF
 10038 DIS =
 10039 ASEL
 10040 LIST
 10041 GOTO UPDAGN
 10042 LABEL ADD
 10043 DIS =
 10044 IF $NOSEL LT 1
 10045  REM *New file, add all.
 10046 IPF EVT.CD1.IPF LN
 10047 ENDIF
 10048 LABEL AGAIN
 10049 IPF EVT.CD2.IPF LN
 10050 DIS =
 10051 CALC $NUM3 = $NOSEL
 10052 RES $RECNO = $NUM3
 10053 IF INSURF LT 0
 10054 IF INEVTR LT 0
 10055 IF INEXDP LT 0
 10056 IF INIEVT LT 0
 10057  ASEL
 10058  GOTO QUERY
 10059 ENDIF
 10060 ENDIF
 10061 ENDIF
 10062 ENDIF
 10063 ASEL
 10064 IPF FMTS.IPF
 10065 LABEL QUERY
 10066 LIST
 10067 DIS 'DO YOU WANT TO ADD CARDS FOR ANOTHER STRESS PERIOD? (Y/N) '
 10068 ACCEPT $CHR3
 10069 IF $CHR3 EQ 'Y'
 10070  GOTO AGAIN
 10071 ENDIF
 10072 LABEL END
 10073 END
```

River Package

The river (RIV) package consists of a control file (RIV.CNTL) and a file or files containing layer, row, column, stage, conductance, altitude of the river bottom, and IKEY. Each simulated stress period can optionally have a new data file for input. The INFO data file that is desired for a particular modeled stress period is specified in item KPER in the RIV.CNTL file. If KPER contains the value RIV.PER2, data file RIV.PER2 will be output. This allows one feature that is not available within the model structure. The model allows use of the data list from

the previous stress period or will read a new list. Output from INFO can use data from any stress period stored for any stress period given. This structure is used for all packages containing list format data by layer, row, and column.

INFO File Structure and Example Data Files

```
 DATAFILE NAME: RIV.CNTL                                          12/5/1988

    1 ITEMS: STARTING IN POSITION    1
 COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  TEXT               30   30  C    -
      **  REDEFINED ITEMS  **
    1  MXRIVR             10   10  I    -
   11  IRIVCB             10   10  I    -
    1  ITMP               10   10  I    -
   11  KPER               10   10  C    -
   29  IKEY                2    2  I    -
    2  CARD               19   19  C    -


 ENTER COMMAND >LI
 $RECNO   TEXT
      1            1        44
      2            1RIV.PER1          9


 DATAFILE NAME: RIV.PER1                                          12/5/1988

    7 ITEMS: STARTING IN POSITION    1
 COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  LAYER              10   10  I    -
   11  ROW                10   10  I    -
   21  COL                10   10  I    -
   31  STAGE              10   10  N    0
   41  COND               10   10  N    8
   51  RBOT               10   10  N    0
   61  IKEY                2    2  I    -
      **  REDEFINED ITEMS  **
    2  CARD               59   59  C    -


 ENTER COMMAND >LI
 $RECNO      LAYER      ROW      COL      STAGE       COND       RBOT IKEY
      1          1        1        3        270 0.00000600        260    9
```

Narrative of River Output Package

1. Select the RIV.CNTL file, open PRIMOS file RIV with INIT, reselect for the first record and output it to PRIMOS file RIV.

2. Loop through each remaining record in RIV.CNTL testing for value .ITMP, corresponding to ITMP in the model, less than zero. If less than or equal to zero, output selects control card only. If greater than zero, use value of item KPER as the file name containing data to output for this stress period, relate to that file, and output data.

Listing of Output Program

```
 10000  PROGRAM SECTION ONE
 10001 REM
 10002 REM *Program: RIVOUT.PG
 10003 REM *This program run from OUTALL if IUNIT(4) > 0
 10004 REM *or from OUTSEL if response was Y(es).
 10005 REM
 10006  FORMAT $NUM13,8,I
 10007  FORMAT $NUM14,8,I
 10008  FORMAT $NUM15,8,I
 10009  FORMAT $NUM30,3,I
 10010  FORMAT $NUM31,3,I
 10011  FORMAT $NUM32,3,I
 10012  FORMAT $CHR16,40,C
 10013  FORMAT $CHR21,40,C
 10014 SEL RIV.CNTL
 10015 OUTPUT RIV INIT
 10016 RES $RECNO = 1
 10017 DIS CARD PRINT
 10018 ASEL
 10019 CA $NUM13 = $NOSEL
 10020 CA $NUM14 = 2
 10021 DO UNTIL $NUM14 GT $NUM13
 10022 RES $RECNO = $NUM14
 10023 IF ITMP LT 0
 10024  DIS CARD PRINT
 10025 ELSE
 10026  DIS CARD PRINT
 10027  CONCAT $CHR16 FROM 'REL ',KPER,' IKEY SEQ'
 10028  EXEC $CHR16
 10029  MOVE 'DIS $1CARD PRINT' TO $CHR21
 10030  NEXT
 10031  EXEC $CHR21
 10032  NEXT OFF
 10033 ENDIF
 10034 CA $NUM15 = $NUM14
 10035 CA $NUM14 = $NUM15 + 1
 10036 ASEL
 10037 DOEND
```

Well Package

The well (WEL) package consists of a control file (WEL.CNTL) and a file or files containing layer, row, column, and discharge. Each file name is entered in item KPER of WEL.CNTL for the appropriate stress period. Files can be output any number of times in any order. The order is controlled by the order of records in the WEL.CNTL file. By this method, if annual pumpage remains constant through time even though two distinct seasons occur, two pumpage data files can be used in an alternating manner. Only those two unique files need to be held in INFO. In this case, the interface output program would alternate between writing each INFO file to the PRIMOS data file WEL.

INFO File Structure and Example Data Files

```
DATAFILE NAME: WEL.CNTL                                          12/5/1988


   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT               40   40  C    -
     **  REDEFINED ITEMS  **
   1  MXWELL             10   10  I    -
  11  IWELCB             10   10  I    -
   1  ITMP               10   10  I    -
  11  KPER               10   10  C    -
  29  IKEY                2    2  I    -
   2  CARD               19   19  C    -

ENTER COMMAND >LI
$RECNO   TEXT
     1            4        42
     2            4WEL.PER1          9


DATAFILE NAME: WEL.PER1                                          12/5/1988

   5 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  LAYER              10   10  I    -
  11  ROW                10   10  I    -
  21  COL                10   10  I    -
  31  Q                  10   10  N    2
  41  IKEY                2    2  I    -
     **  REDEFINED ITEMS  **
   2  CARD               39   39  C    -

ENTER COMMAND >LI
$RECNO        LAYER        ROW        COL          Q IKEY
     1            1          1          2       0.50  9
     2            2          2          4       0.75  9
     3            3          1          5       0.50  9
     4            4          2          4       0.75  9
```

Narrative of Well Output Program

1. Select the WEL.CNTL file, open PRIMOS file WEL with INIT, reselect for the first record and output to PRIMOS file WEL.

2. Loop through each remaining record in WEL.CNTL testing for value ITMP less than, equal to or greater than zero. If less than or equal to zero, output selected control card only. If greater than zero, use value of item KPER as the file name containing data to output for this stress period, relate to that file and output data.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: WELOUT.PG
10003 REM *This program run from OUTALL if IUNIT(2) > 0
10004 REM *or from OUTSEL if response was Y(es).
10005 REM
10006  FORMAT $NUM13,8,I
10007  FORMAT $NUM14,8,I
10008  FORMAT $NUM15,8,I
10009  FORMAT $NUM30,3,I
10010  FORMAT $NUM31,3,I
10011  FORMAT $NUM32,3,I
10012  FORMAT $CHR16,40,C
10013  FORMAT $CHR21,40,C
10014 SEL WEL.CNTL
10015 OUTPUT WEL INIT
10016 RES $RECNO = 1
10017 DIS CARD PRINT
10018 ASEL
10019 CA $NUM13 = $NOSEL
10020 CA $NUM14 = 2
10021 DO UNTIL $NUM14 GT $NUM13
10022 RES $RECNO = $NUM14
10023 IF ITMP LT 0
10024  DIS CARD PRINT
10025 ELSE
10026  DIS CARD PRINT
10027  CONCAT $CHR16 FROM 'REL ',KPER,' IKEY SEQ'
10028  EXEC $CHR16
10029  MOVE 'DIS $1CARD PRINT' TO $CHR21
10030  NEXT
10031  EXEC $CHR21
10032  NEXT OFF
10033 ENDIF
10034 CA $NUM15 = $NUM14
10035 CA $NUM14 = $NUM15 + 1
10036 ASEL
10037 DOEND
```

Drain Package

The drain (DRN) package consists of a control file (DRV.CNTL) and a file or files containing layer, row, column, elevation, conductance, and IKEY. Each simulated stress period can optionally have a new data file for input. The INFO data file that is desired for a particular modeled stress period is specified in item KPER in the DRN.CNTL file. If KPER contains the value DRN.PER2, data file DRN.PER2 will be output.

INFO File Structure and Example Data Files

```
DATAFILE NAME: DRN.CNTL                                          12/5/1988


   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT                30   30  C    -
     **  REDEFINED ITEMS  **
   1  MXDRN               10   10  I    -
  11  IDRNCB              10   10  I    -
   1  ITMP                10   10  I    -
  11  KPER                10   10  C    -
  29  IKEY                 2    2  I    -
   2  CARD                19   19  C    -

ENTER COMMAND >LI
$RECNO   TEXT
     1            2        43
     2            2DRN.PER1          9


DATAFILE NAME: DRN.PER1                                          12/5/1988


   6 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  LAYER               10   10  I    -
  11  ROW                 10   10  I    -
  21  COL                 10   10  I    -
  31  ELEVATION           10   10  N    0
  41  COND                10   10  N    7
  51  IKEY                 2    2  I    -
     **  REDEFINED ITEMS  **
   2  CARD                49   49  C    -


ENTER COMMAND >LI
$RECNO        LAYER        ROW         COL  ELEVATION       COND IKEY
     1            1          1           1        250  0.2000000  9
     2            1          2           1        250  0.2000000  9
```

Narrative of Drain Output Program

1. Select the DRN.CNTL file, open PRIMOS file DRN with INIT, reselect for the first record and output it to PRIMOS file DRN.

2. Loop through each remaining record in DRN.CNTL testing for value ITMP less than, equal to or greater than zero. If less than or equal to zero, output selected control card only. If greater than zero, use value of item KPER as the file name containing data to output for this stress period, relate to that file and output data.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: DRNOUT.PG
10003 REM *This program run from OUTALL if IUNIT(3) > 0
10004 REM *or from OUTSEL if response was Y(es).
10005 REM
10006  FORMAT $NUM13,8,I
10007  FORMAT $NUM14,8,I
10008  FORMAT $NUM15,8,I
10009  FORMAT $NUM30,3,I
10010  FORMAT $NUM31,3,I
10011  FORMAT $NUM32,3,I
10012  FORMAT $CHR16,40,C
10013  FORMAT $CHR21,40,C
10014 SEL DRN.CNTL
10015 OUTPUT DRN INIT
10016 RES $RECNO = 1
10017 DIS CARD PRINT
10018 ASEL
10019 CA $NUM13 = $NOSEL
10020 CA $NUM14 = 2
10021 DO UNTIL $NUM14 GT $NUM13
10022 RES $RECNO = $NUM14
10023 IF ITMP LT 0
10024  DIS CARD PRINT
10025 ELSE
10026  DIS CARD PRINT
10027  CONCAT $CHR16 FROM 'REL ',KPER,' IKEY SEQ'
10028  EXEC $CHR16
10029  MOVE 'DIS $1CARD PRINT' TO $CHR21
10030  NEXT
10031  EXEC $CHR21
10032  NEXT OFF
10033 ENDIF
10034 CA $NUM15 = $NUM14
10035 CA $NUM14 = $NUM15 + 1
10036 ASEL
10037 DOEND
```

General-Head Boundary Package

The general-head boundary (GHB) package consists of a control file (GHB.CNTL) and a file or files containing layer, row, column, boundary head, conductance, and IKEY. Each simulated stress period can optionally have a new data file for input. The INFO data file that is desired for a particular modeled stress period is specified in item KPER in the GHB.CNTL file. If KPER contains the value GHB.PER2, data file GHB.PER2 will be output.

INFO File Structure

```
DATAFILE NAME: GHB.CNTL                                          12/5/1988


   1 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  TEXT               30   30  C    -
     **  REDEFINED ITEMS  **
   1  MXBND              10   10  I    -
  11  IGHBCB             10   10  I    -
   1  ITMP               10   10  I    -
  11  KPER               10   10  C    -
  29  IKEY                2    2  I    -
   2  CARD               19   19  C    -

ENTER COMMAND >LI
$RECNO   TEXT
     1            2        46
     2            2GHB.PER1           9


DATAFILE NAME: GHB.PER1                                          12/5/1988


   6 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
   1  LAYER              10   10  I    -
  11  ROW                10   10  I    -
  21  COL                10   10  I    -
  31  B.HEAD             10   10  N    0
  41  COND               10   10  N    7
  51  IKEY                2    2  I    -
     **  REDEFINED ITEMS  **
   2  CARD               49   49  C    -

$RECNO        LAYER         ROW         COL      B.HEAD       COND IKEY
     1            1           1           1     270.000    0.00010  9
     2            1           1           2     270.000    0.00010  9
```

Narrative of General-Head Boundary Output Program

1. Select the GHB.CNTL file, open PRIMOS file GHB with INIT, reselect for the first record and output it to PRIMOS file GHB.

2. Loop through each remaining record in GHB.CNTL testing for value ITMP less than, equal to, or greater than zero. If less than or equal to zero, output selected control card only. If greater than zero, use value of item KPER as the file name containing data to output for this stress period, relate to that file and output data.

Listing of Output Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: GHBOUT.PG
10003 REM *This program run from OUTALL if IUNIT(7) > 0
10004 REM *or from OUTSEL if response was Y(es).
10005 REM
10006  FORMAT $NUM13,8,I
10007  FORMAT $NUM14,8,I
10008  FORMAT $NUM15,8,I
10009  FORMAT $NUM30,3,I
10010  FORMAT $NUM31,3,I
10011  FORMAT $NUM32,3,I
10012  FORMAT $CHR16,40,C
10013  FORMAT $CHR21,40,C
10014 SEL GHB.CNTL
10015 OUTPUT GHB INIT
10016 RES $RECNO = 1
10017 DIS CARD PRINT
10018 ASEL
10019 CA $NUM13 = $NOSEL
10020 CA $NUM14 = 2
10021 DO UNTIL $NUM14 GT $NUM13
10022 RES $RECNO = $NUM14
10023 IF ITMP LT 0
10024  DIS CARD PRINT
10025 ELSE
10026  DIS CARD PRINT
10027  CONCAT $CHR16 FROM 'REL ',KPER,' IKEY SEQ'
10028  EXEC $CHR16
10029  MOVE 'DIS $1CARD PRINT' TO $CHR21
10030  NEXT
10031  EXEC $CHR21
10032  NEXT OFF
10033 ENDIF
10034 CA $NUM15 = $NUM14
10035 CA $NUM14 = $NUM15 + 1
10036 ASEL
10037 DOEND
```

Strongly Implicit Procedure, Slice-Successive Overrelaxation, and Output Control Packages

Strongly implicit procedure (SIP), slice-successive overrelaxation (SOR), and output control (OC) packages can only contain controlling information. The INFO file structures and output programs are all similar, and in each case, only one INFO data file is defined—SIP.CNTL, SOR.CNTL, or OC.CNTL. Output is produced by selecting that file and issuing the "SAVE COMPRESS" command that moves the file directly to PRIMOS. Those files match the PRIMOS files exactly and, therefore, can be moved from PRIMOS into INFO by issuing the "GET filename COPY" command from within INFO.

INFO File Structure and Example Data Files

```
 DATAFILE NAME: SIP.CNTL                                          12/5/1988

    1 ITEMS: STARTING IN POSITION    1
 COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  TEXT               60   60  C    -
      **  REDEFINED ITEMS  **
    1  MXITER             10   10  I    -
   11  NPARM              10   10  I    -
    1  ACCL               10   10  N    3
   11  HCLOSE             10   10  N    6
   21  IPCALC             10   10  I    -
   31  WSEED              10   10  N    6
   41  IPRSIP             10   10  I    -

 ENTER COMMAND >LI
 $RECNO   TEXT
      1           78         5
      2        1.200    .00002         1    .00002        91
```

```
 DATAFILE NAME: SOR.CNTL                                          12/5/1988

    1 ITEMS: STARTING IN POSITION    1
 COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  TEXT               40   40  C    -
      **  REDEFINED ITEMS  **
    1  MXITER             10   10  I    -
    1  ACCL               10   10  N    3
   11  HCLOSE             10   10  N    6
   21  IPRSOR             10   10  I    -
```

```
 DATAFILE NAME: OC.CNTL                                          12/5/1988


    1 ITEMS: STARTING IN POSITION    1
 COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  TEXT                50   50  C    -
      **  REDEFINED ITEMS  **
    1  IHEDFM              10   10  I    -
   11  IDDNFM              10   10  I    -
   21  IHEDUN              10   10  I    -
   31  IDDNUN              10   10  I    -
    1  INCODE              10   10  I    -
   11  IHDDFL              10   10  I    -
   21  IBUDFL              10   10  I    -
   31  ICBCFL              10   10  I    -
    1  HDPR                10   10  I    -
   11  DDPR                10   10  I    -
   21  HDSV                10   10  I    -
   31  DDSV                10   10  I    -


 ENTER COMMAND >LI
 $RECNO   TEXT
      1            3         3        48        49
      2            0         1         1         1
      3            1         1         1         1
```

Listing of Strongly Implicit Procedure Output Program

```
 10000  PROGRAM SECTION ONE
 10001 REM
 10002 REM *This program is run either from OUTALL or OUTSEL
 10003 REM *to output necessary data for the MacDonald-Harbaugh
 10004 REM *modular model.
 10005 REM
 10006 SEL SIP.CNTL
 10007 SAVE SIP COMPRESS INIT
```

Listing of Slice-Successive Overrelaxation Output Program

```
 10000  PROGRAM SECTION ONE
 10001 REM
 10002 REM *This program is run either from OUTALL or OUTSEL
 10003 REM *to output necessary data for the MacDonald-Harbaugh
 10004 REM *modular model.
 10005 REM
 10006 SEL SOR.CNTL
 10007 SAVE SOR COMPRESS INIT
```

Listing of Output Control Program

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *This program is run either from OUTALL or OUTSEL
10003 REM *to output necessary data for the MacDonald-Harbaugh
10004 REM *modular model.
10005 REM
10006 SEL OC.CNTL
10007 SAVE OC COMPRESS INIT
```

Control of the Package Output Programs

Two INFO programs—OUTALL and OUTSEL—and the MAIN.CPL program control the output programs. The INFO programs are similar to main programs in Fortran and contain the package output programs as subroutines. Running either OUTALL or OUTSEL depends on the options chosen when running MAIN.CPL.

Usage and argument format used within the interface CPLs are similar to commands used in ARC/INFO. If the user wishes to see the usage, type 'R MAIN' and the usage, which includes arguments will be displayed. That usage and a description of options follows.

Usage: R MAIN <INFO|PRIMOS|BOTH> <LIST|WIDE> <ARC|ASCII>

1. <INFO|PRIMOS|BOTH>--The argument controls output from INFO to the model and is a required argument as are all arguments bracketed by the greater than (>) and less than (<) symbols. The flow of MAIN.CPL is affected by the choice of option. The effect of the choice of each option is as follows:

   a. INFO--Generates a complete new set of PRIMOS level package data files for the model by entering INFO and running OUTALL.

   b. PRIMOS--Runs the model with the existing PRIMOS data files. In this case, neither OUTALL nor OUTSEL are run, and none of the package output programs are run.

   c. BOTH--Runs the model with some of the existing PRIMOS data files and with some new files created from INFO. MAIN.CPL will enter INFO and run INFO program OUTSEL.IF, which allows for full-screen updating of OUTSEL.DF. OUTSEL.DF contains one record for each package with a "Yes" or "No" flag. "Yes" means that the package output program will be run and will overwrite the existing PRIMOS data file with data currently in INFO. "No" means that the package output program will not be run.

2. <LIST|WIDE>--Controls the format with which two-dimensional arrays are written to the PRIMOS package data files from INFO.

   a. LIST--Outputs all two-dimensional arrays as lists with one value per line.

   b. WIDE--Outputs all two-dimensional arrays across the page by rows of the model grid. This format is used for the sample files shown by McDonald and Harbaugh (1988).

3. <ARC|ASCII> This option specifies whether model output will be returned to ARC/INFO.

   a. ARC--Generates PRIMOS data files of selected model outputs for entry to ARC/INFO. Following the model run, MAIN.CPL will run a Fortran program called WLSIN.F77 to generate those files.

   b. ASCII--Terminates processing following the model run. ASCII means only standard model outputs are desired and no ARC coverages will be generated. Note that if the PRIMOS option and the ASCII option are selected, the result will be simply to run the model with no interaction with ARC/INFO.

Program MAIN.CPL will write a PRIMOS como file that will save everything that appears on screen from the time that the como file is opened until it is closed. The como file is a means of tracking program execution and can be used to locate errors in programming if they occur.

Listing of MAIN.CPL Program

```
/*************************************************************
/* PROGRAM NAME: MAIN.CPL
/*
/* DESCRIPTION: This CPL will run the GW model using
/*   inputs from ARC/INFO (INFO option), the existing
/*   Primos files (PRIMOS option), or a combination of
/*   both (BOTH option, specific files picked from a
/*   menu).  Arrays can be output to Primos as lists or in
/*   wide format (written across page).  List vs. wide format
/*   CPU and IO are about equal, if less than 132 characters
/*   needed per row (ie. format times ncols < 132). If > 132
/*   characters, wide takes at least 2 to 3 times more
/*   computer time.
/*        To move model output into ARC, select
/*   ARC output option.  Either WLSIN.CPL or CBCIN.CPL
/*   can then be run to generate coverages of specified
/*   output parameters.
/*
/* AUTHOR: Peter Van Metre                  DATE:4/27/87
/*************************************************************
```

```
COMO COMO.MAIN
TIME
&ARGS INPUT ; FMTIN ; OUTPUT

/* Test if first two inputs have been entered.
&IF [NULL %OUTPUT%] &THEN &DO
   TYPE Usage: R MAIN <INFO|PRIMOS|BOTH> <LIST|WIDE> <ARC|ASCII>
   &RETURN 1
&END

&IF %INPUT% = INFO | %INPUT% = PRIMOS | %INPUT% = BOTH &THEN &GOTO %INPUT%

&LABEL INFO
&IF %FMTIN% = WIDE | %FMTIN% = LIST &THEN &GOTO %FMTIN%

&LABEL WIDE
&DATA ARC INFO
  RUN OUTALL
  Q STOP
&END
&GOTO PRIMOS
&LABEL LIST
&DATA ARC INFO
  FORMAT $CHR39,4,C
  MOVE 'LIST' TO $CHR39
  RUN OUTALL
  Q STOP
&END
&GOTO PRIMOS

&LABEL BOTH
&DATA ARC INFO
SEL OUTSEL.DF
PURG
Y
IPF OUTSEL.IF LN
&TTY
&END

&IF %FMTIN% = WIDE | %FMTIN% = LIST &THEN &GOTO %FMTIN%2


&LABEL WIDE2
&DATA ARC INFO
  RUN OUTSEL
  Q STOP
&END
&GOTO PRIMOS

&LABEL LIST2
&DATA ARC INFO
  FORMAT $CHR39,4,C
  MOVE 'LIST' TO $CHR39
  RUN OUTSEL
  Q STOP
```

```
&END
&GOTO PRIMOS
&LABEL PRIMOS
/* SEG MODEL[2] (Run CPL Program SEGMODEL.CPL)
&DATA R SEGMODEL
&END


&IF %OUTPUT% = ARC | %OUTPUT% = ASCII &THEN &GOTO %OUTPUT%
&LABEL ARC
A *>INFO
R WLSIN
UP


&LABEL ASCII
TIME
COMO -E
```

---

[2]Listing of CPL Program SEGMODEL.CPL
(called from MAIN.CPL)

```
/*********************************************************
/* PROGRAM NAME: SEGMODEL.CPL
/*   Unit numbers correspond to FORTRAN unit numbers in
/*   INFO data file BAS.CNTL (IUNITs).  Units not in use
/*   should be commented out with '/*' in first two spaces.
/*   Files ending in 'UNF' are unformatted files
/*   for model outputs 'saved on disk'; unit numbers must
/*   match IxxxCB value for each package when output is
/*   saved.
/* DESCRIPTION: Called from MAIN.CPL to run the GW model.
/*********************************************************
TIME
FOPEN *>INFO>BAS 5 1
FOPEN *>INFO>INOUT 6 2
FOPEN *>INFO>BCF 29 1
FOPEN *>INFO>WEL 30 1
FOPEN *>INFO>DRN 31 1
FOPEN *>INFO>RIV 32 1
FOPEN *>INFO>EVT 33 1
FOPEN *>INFO>GHB 35 1
FOPEN *>INFO>RCH 36 1
FOPEN *>INFO>SIP 37 1
FOPEN *>INFO>OC 38 1
FOPEN *>INFO>BCFUNF 41 2
FOPEN *>INFO>WELUNF 42 2
FOPEN *>INFO>DRNUNF 43 2
FOPEN *>INFO>RIVUNF 44 2
FOPEN *>INFO>EVTUNF 45 2
FOPEN *>INFO>GHBUNF 46 2
FOPEN *>INFO>RCHUNF 47 2
FOPEN *>INFO>WLSUNF 48 2
FOPEN *>INFO>DDSUNF 49 2
```

```
/* Pathname must point to SEG directory for GW model on users system.
/* If using BIND, change SEG to R.
&DATA SEG MODELS>GW>MODFLOW
&END
```

Description of Programs OUTALL and OUTSEL

INFO program OUTALL will run all of the package output programs for which IUNIT in BAS.CNTL is greater than zero. This includes all packages being used in the particular model and creates a complete new set of PRIMOS data files for the model. OUTALL probably would be run only to create the files for a first model run or after some major update such as changing the model grid.

INFO program OUTSEL will run only those package output programs for which a "Yes" response is entered in the INFO data file OUTSEL.DF and for which IUNIT in BAS.CNTL is greater than zero. OUTSEL.DF is automatically brought on screen for menu-driven updating by MAIN.CPL if the "BOTH" option is specified. OUTSEL would be run after one or more model data files were updated in INFO to create one or more PRIMOS data files for the model.

Narrative of INFO Program OUTALL

1. Select INFO data file BAS.CNTL, reselect for record number four that contains the IUNIT values for all other packages, and read those values into dimensioned INFO variables. Reselect for record number three and move number of layers, rows, and columns into INFO variables for later use in controlling output.

2. Test each IUNIT value and run each package output program for which the corresponding IUNIT value is greater than zero.

Listing of INFO Program OUTALL

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: OUTALL
10003 REM *Reads BAS.CNTL and outputs all packages for which
10004 REM *IUNIT is > 0. Run from INFO or called by MAIN.CPL
10005 REM *if input option = INFO.
10006 REM
10007 SEL BAS.CNTL
10008  FORMAT $NUM1,3,I
10009  FORMAT $NUM2,3,I
10010  FORMAT $NUM3,3,I
10011  FORMAT $NUM4,3,I
```

```
 10012  FORMAT $NUM5,3,I
 10013  FORMAT $NUM6,3,I
 10014  FORMAT $NUM7,3,I
 10015  FORMAT $NUM8,3,I
 10016  FORMAT $NUM9,3,I
 10017  FORMAT $NUM10,3,I
 10018  FORMAT $NUM11,3,I
 10019  FORMAT $NUM13,8,I
 10020  FORMAT $NUM14,8,I
 10021  FORMAT $NUM15,8,I
 10022  FORMAT $NUM30,3,I
 10023  FORMAT $NUM31,3,I
 10024  FORMAT $NUM32,3,I
 10025  FORMAT $NUM33,3,I
 10026  FORMAT $NUM34,3,I
 10027  FORMAT $NUM35,3,I
 10028  FORMAT $NUM36,3,I
 10029  FORMAT $NUM37,3,I
 10030  FORMAT $NUM38,3,I
 10031  FORMAT $CHR16,40,C
 10032  FORMAT $CHR21,40,C
 10033  FORMAT $CHR26,16,C
 10034  FORMAT $CHR39,4,C
 10035 RES BY $RECNO EQ 3
 10036 CALC $NUM37 = NCOL
 10037 CALC $NUM36 = NROW
 10038 ASEL
 10039 RES BY $RECNO EQ 4
 10040 REM *BCF
 10041  CA $NUM1 = IUNIT1
 10042 REM *WEL
 10043  CA $NUM2 = IUNIT2
 10044 REM *DRN
 10045  CA $NUM3 = IUNIT3
 10046 REM *RIV
 10047  CA $NUM4 = IUNIT4
 10048 REM *EVT
 10049  CA $NUM5 = IUNIT5
 10050 REM *GHB
 10051  CA $NUM6 = IUNIT6
 10052 REM *RCH
 10053  CA $NUM7 = IUNIT8
 10054 REM *SIP
 10055  CA $NUM8 = IUNIT9
 10056 REM *SOR
 10057  CA $NUM9 = IUNIT11
 10058 REM *OUT
 10059  CA $NUM10 = IUNIT12
 10060 REM *CHD
 10061  CA $NUM11 = IUNIT20
 10062 REM
 10063 REM *Now run XXXout programs
 10064 REM
 10065  RUN BASOUT.PG LINK
 10066 IF $NUM1 GT 0
```

```
10067  RUN BCFOUT.PG LINK
10068 ENDIF
10069 IF $NUM2 GT 0
10070  RUN WELOUT.PG LINK
10071 ENDIF
10072 IF $NUM3 GT 0
10073  RUN DRNOUT.PG LINK
10074 ENDIF
10075 IF $NUM4 GT 0
10076  RUN RIVOUT.PG LINK
10077 ENDIF
10078 IF $NUM5 GT 0
10079  RUN EVTOUT.PG LINK
10080 ENDIF
10081 IF $NUM6 GT 0
10082  RUN GHBOUT.PG LINK
10083 ENDIF
10084 IF $NUM7 GT 0
10085  RUN RCHOUT.PG LINK
10086 ENDIF
10087 IF $NUM8 GT 0
10088  RUN SIPOUT.PG LINK
10089 ENDIF
10090 IF $NUM9 GT 0
10091  RUN SOROUT.PG LINK
10092 ENDIF
10093 IF $NUM10 GT 0
10094  RUN OCOUT.PG LINK
10095 ENDIF
10096 IF $NUM11 GT 0
10097  RUN CHDOUT.PG LINK
10098 ENDIF
10099 END
```

Narrative of INFO Program OUTSEL

1. Select INFO data file BAS.CNTL, reselect for record number four which contains the IUNIT values for all other packages, and read those values into dimensioned INFO variables. Reselect for record number three and move number of layers, rows, and columns into INFO variables for later use in controlling output.

2. Select INFO data file OUTSEL.DF and read the "Yes" or "No" response into dimensioned INFO variables.

3. Test each variable for value equal Y or y and IUNIT value greater than zero and run each package output program passing those two tests.

Listing of INFO Program OUTSEL

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: OUTSEL
10003 REM *Reads BAS.CNTL and OUTSEL.DF outputs all packages for
10004 REM *which IUNIT is > 0 and entry in OUTSEL.DF is yes (Y or y).
10005 REM
10006 SEL BAS.CNTL
10007  FORMAT $NUM1,3,I
10008  FORMAT $NUM2,3,I
10009  FORMAT $NUM3,3,I
10010  FORMAT $NUM4,3,I
10011  FORMAT $NUM5,3,I
10012  FORMAT $NUM6,3,I
10013  FORMAT $NUM7,3,I
10014  FORMAT $NUM8,3,I
10015  FORMAT $NUM9,3,I
10016  FORMAT $NUM10,3,I
10017  FORMAT $NUM13,8,I
10018  FORMAT $NUM14,8,I
10019  FORMAT $NUM15,8,I
10020  FORMAT $NUM30,3,I
10021  FORMAT $NUM31,3,I
10022  FORMAT $NUM32,3,I
10023  FORMAT $NUM33,3,I
10024  FORMAT $NUM34,3,I
10025  FORMAT $NUM35,3,I
10026  FORMAT $NUM36,3,I
10027  FORMAT $NUM37,3,I
10028  FORMAT $NUM38,3,I
10029  FORMAT $CHR16,40,C
10030  FORMAT $CHR21,40,C
10031  FORMAT $CHR26,16,C
10032  FORMAT $CHR39,4,C
10033 RES BY $RECNO EQ 3
10034 CALC $NUM37 = NCOL
10035 CALC $NUM36 = NROW
10036 ASEL
10037 RES BY $RECNO EQ 4
10038 REM *BCF
10039  CA $NUM1 = IUNIT1
10040 REM *WEL
10041  CA $NUM2 = IUNIT2
10042 REM *DRN
10043  CA $NUM3 = IUNIT3
10044 REM *RIV
10045  CA $NUM4 = IUNIT4
10046 REM *EVT
10047  CA $NUM5 = IUNIT5
10048 REM *GHB
10049  CA $NUM6 = IUNIT6
10050 REM *RCH
10051  CA $NUM7 = IUNIT8
```

```
10052 REM *SIP
10053  CA $NUM8 = IUNIT9
10054 REM *SOR
10055  CA $NUM9 = IUNIT11
10056 REM *OUT
10057  CA $NUM10 = IUNIT12
10058 REM
10059 REM *Now read OUTSEL.DF to determine which files to output.
10060 REM
10061 SEL OUTSEL.DF
10062 IF BCF NE 'Y' AND BCF NE 'y'
10063  CA $NUM1 = 0
10064 ENDIF
10065 IF WEL NE 'Y' AND WEL NE 'y'
10066  CA $NUM2 = 0
10067 ENDIF
10068 IF DRN NE 'Y' AND DRN NE 'y'
10069  CA $NUM3 = 0
10070 ENDIF
10071 IF RIV NE 'Y' AND RIV NE 'y'
10072  CA $NUM4 = 0
10073 ENDIF
10074 IF EVT NE 'Y' AND EVT NE 'y'
10075  CA $NUM5 = 0
10076 ENDIF
10077 IF GHB NE 'Y' AND GHB NE 'y'
10078  CA $NUM6 = 0
10079 ENDIF
10080 IF RCH NE 'Y' AND RCH NE 'y'
10081  CA $NUM7 = 0
10082 ENDIF
10083 IF SIP NE 'Y' AND SIP NE 'y'
10084  CA $NUM8 = 0
10085 ENDIF
10086 IF SOR NE 'Y' AND SOR NE 'y'
10087  CA $NUM9 = 0
10088 ENDIF
10089 IF OC NE 'Y' AND OC NE 'y'
10090  CA $NUM10 = 0
10091 ENDIF
10092 REM
10093 REM *Now run XXXout programs
10094 REM
10095 IF BAS EQ 'Y' OR BAS EQ 'y'
10096  RUN BASOUT.PG LINK
10097 ENDIF
10098 IF $NUM1 GT 0
10099  RUN BCFOUT.PG LINK
10100 ENDIF
10101 IF $NUM2 GT 0
10102  RUN WELOUT.PG LINK
10103 ENDIF
10104 IF $NUM3 GT 0
10105  RUN DRNOUT.PG LINK
10106 ENDIF
```

```
10107 IF $NUM4 GT 0
10108  RUN RIVOUT.PG LINK
10109 ENDIF
10110 IF $NUM5 GT 0
10111  RUN EVTOUT.PG LINK
10112 ENDIF
10113 IF $NUM6 GT 0
10114  RUN GHBOUT.PG LINK
10115 ENDIF
10116 IF $NUM7 GT 0
10117  RUN RCHOUT.PG LINK
10118 ENDIF
10119 IF $NUM8 GT 0
10120  RUN SIPOUT.PG LINK
10121 ENDIF
10122 IF $NUM9 GT 0
10123  RUN SOROUT.PG LINK
10124 ENDIF
10125 IF $NUM10 GT 0
10126  RUN OCOUT.PG LINK
10127 ENDIF
10128 END
```

Listing of INFO Input Form OUTSEL.DF

```
ENTER COMMAND >TYPE OUTSEL.IF
FORM NAME: OUTSEL.IF                                          12/9/1988
2,25,'SELECTED OUTPUT FROM ARC/INFO'
3,30,'FOR GW MODEL RUN'
5,28,'PACKAGE   OUTPUT?(Y/N)'
6,30,'BAS'
6,40,':'
7,30,'BCF'
7,40,':'
8,30,'WEL'
8,40,':'
9,30,'DRN'
9,40,':'
10,30,'RIV'
10,40,':'
11,30,'EVT'
11,40,':'
12,30,'GHB'
12,40,':'
13,30,'RCH'
13,40,':'
14,30,'SIP'
14,40,':'
15,30,'SOR'
15,40,':'
16,30,'OC'
16,40,':'
18,2,'IF EXECUTED FROM MAIN.CPL, ENTER "Q STOP" AT "ENTER COMMAND>"
```

```
PROMPT'
 6,41,BAS
 7,41,BCF
 8,41,WEL
 9,41,DRN
 10,41,RIV
 11,41,EVT
 12,41,GHB
 13,41,RCH
 14,41,SIP
 15,41,SOR
 16,41,OC
```

INFO File Structure and Example Data Files for OUTSEL.DF

```
 DATAFILE NAME: OUTSEL.DF                                        12/9/1988

   11 ITEMS: STARTING IN POSITION    1
 COL  ITEM NAME         WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  BAS                1    1  C    -
    2  BCF                1    1  C    -
    3  WEL                1    1  C    -
    4  DRN                1    1  C    -
    5  RIV                1    1  C    -
    6  EVT                1    1  C    -
    7  GHB                1    1  C    -
    8  RCH                1    1  C    -
    9  SIP                1    1  C    -
   10  SOR                1    1  C    -
   11  OC                 1    1  C    -

 ENTER COMMAND >LI
 $RECNO   BAS BCF WEL DRN RIV EVT GHB RCH SIP SOR OC
      1   Y   Y   N   N   N   N   N   N   N   N   N
```

Returning Model Output to ARC/INFO

Three CPL programs—MAIN.CPL, CBCIN.CPL and WLSIN.CPL—affect the return of model-output heads, drawdowns, and flow terms to ARC/INFO by using one Fortran program and two INFO programs. WLSIN.CPL and CBCIN.CPL can be run at any time after a model run has been completed if MAIN.CPL was run with the ARC option. WLSIN.CPL and CBCIN.CPL will create a coverage of only one model output data set each time they are run—for example, head values for all layers for a particular stress period and time step.

The process of returning model outputs to ARC/INFO is accomplished by the following steps.

1. If the ARC option is specified when MAIN.CPL is run, MAIN.CPL will run Fortran program WLSIN.F77. This program

will open the output control (OC) PRIMOS file and the appropriate PRIMOS package data files as necessary, determine what was saved on disk, then read and reformat those records for entry to INFO. Each reformatted data set will carry a unique PRIMOS filename assigned by the program; for example, heads for stress period 2 after time step 1 will be called (by default) HED.02.01.

2. At any time after a model run is completed, the user can run either CBCIN.CPL or WLSIN.CPL. Either requires one argument, which is the PRIMOS filename of the reformatted (step 1) data set awaiting entry to INFO. Each CPL follows the same general steps which are:

   a. Determine if a coverage already exists by the data set name given.

   b. If that coverage does exist, enter INFO and run INFO program LOADWLS.PG. LOADWLS.PG will bring the data set into INFO using "GET filename COPY," relate that data set to the .PAT file of the same name and move that data into that file.

   c. If that coverage does not exist, use ARC commands to make a copy of the NODES coverage, and add the necessary INFO items. Then enter INFO and run LOADWLS.PG to bring the specified data set into INFO and load those values to the new coverage.

Listing of CPL Program WLSIN.CPL

```
/*****************************************************************
/* PROGRAM NAME: WLSIN.CPL
/*
/* DESCRIPTION:  This cpl will load the requested model output,
/*   which was saved unformatted, then reformatted for INFO
/*   by WLSIN.F77 (eg. HED.03.01=heads for stress period 3,
/*   time step 1) into ARC/INFO.  The ARC coverage name will
/*   be the same as the PRIMOS file name (HED.03.01 for the
/*   example).
/*
/* SYNTAX 'R WLSIN <file.name>
/* DEFAULT FILE LOCATION IS IN *>INFO DIR
/*
/*****************************************************************
&ARGS INFILE
/*Test if usage correct
&IF [NULL %INFILE%] &THEN &DO
   TYPE Usage: R WLSIN <primos file.name>
   &RETURN 1
&END
/*Test if coverage exists
&IF [EXISTS %INFILE% -DIR] &THEN &GOTO LAB
```

```
&DATA ARC COPY NODES %INFILE%
&END

&DATA ARC INFO
SEL WLS.DAT
PURGE
Y
/*Model output pathname, if not in INFO directory
/*used by ARC, should be entered here in the 'GET-COPY'
/*command.
GET %INFILE% COPY
Q STOP
&END
&DATA ARC JOINITEM %INFILE%.PAT WLS.DAT %INFILE%.PAT KEY KEY LINK
&END
&RETURN

/*If coverage did exist, enter here
&LABEL LAB
&DATA ARC INFO
SEL WLS.DAT
PURGE
Y
GET %INFILE% COPY
SEL %INFILE%.PAT
REL WLS.DAT BY KEY LINK
RUN LOADWLS.PG
/* LOADWLS.PG is an INFO program containing one calculate statement
/* for each layer in model.  It will need to be modified for each
/* application, depending on number of layers simulated.
Q STOP
&END
&RETURN
```

Listing of CPL Program CBCIN.CPL

```
/*****************************************************************
/* PROGRAM NAME: CBCIN.CPL
/*
/* DESCRIPTION:  This cpl will load the requested model output,
/*   which was saved unformatted, then reformatted for INFO
/*   by CBCIN.F77 (eg. RIV.03.01=river node flows for stress
/*   time step 1) into ARC/INFO.  The ARC coverage name will
/*   be the same as the PRIMOS file name (RIV03.01 for the
/*   example).
/*
/* SYNTAX 'R CBCIN <file.name>
/* DEFAULT FILE LOCATION IS IN *>INFO DIR
/*
/*****************************************************************
&ARGS INFILE
/*Test if usage correct
&IF [NULL %INFILE%] &THEN &DO
```

```
   TYPE Usage: R CBCIN <primos file.name>
   &RETURN 1
&END

/*Test if coverage exists
&IF [EXISTS %INFILE% -DIR] &THEN &GOTO LAB

&DATA ARC COPY NODES %INFILE%
&END
/*Default model output pathname, if not in INFO directory
/*used by ARC should be entered here in the 'GET-COPY'
/*command

&DATA ARC INFO
SEL CBC.DAT
PURGE
Y
GET %INFILE% COPY
Q STOP
&END
&DATA ARC JOINITEM %INFILE%.PAT CBC.DAT %INFILE%.PAT KEY KEY LINK
&END
&RETURN



/*If coverage did exist, enter here
&LABEL LAB
&DATA ARC INFO
SEL CBC.DAT
PURGE
Y
GET %INFILE% COPY
SEL %INFILE%.PAT
REL CBC.DAT BY KEY LINK
RUN LOADWLS.PG
/* LOADWLS.PG is an INFO program containing one calculate statement
/* for each layer in model.  It will need to be modified for each
/* application, depending on number of layers.
Q STOP
&END
&RETURN
```

INFO File Structure of Files CBC.DAT and WLS.DAT

```
 DATAFILE NAME: CBC.DAT

    4 ITEMS: STARTING IN POSITION    1
 COL   ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
    1  LAY1                12   12  N    5
   13  LAY2                12   12  N    5
   25  LAY3                12   12  N    5
   37  LAY4                12   12  N    5
```

```
ENTER COMMAND >LI
```

$RECNO	LAY1	LAY2	LAY3	LAY4
1	-0.31440	-0.10342	-0.07110	-0.05892
2	-0.03636	-0.08400	-0.09931	-0.09921
3	0.00000	-0.13171	-0.11544	-0.11637
4	0.00000	-0.03337	-0.08699	-0.01782
5	0.00000	0.00000	0.11766	0.03803
6	0.00000	0.00000	-0.00540	-0.00214
7	0.00000	0.00000	0.00000	-0.04041
8	0.00000	0.00000	0.00000	-0.05003
9	0.00000	0.00000	0.00000	0.00000
10	-0.21430	-0.09512	-0.07336	-0.06501
11	-0.10019	-0.11520	-0.11911	-0.12887
12	0.00000	-0.24951	-0.15375	-0.24306
13	0.00000	0.11479	0.02516	0.12596
14	0.00000	0.00000	0.05881	0.05081
15	0.00000	0.00000	-0.01357	-0.00159
16	0.00000	0.00000	0.00000	-0.04082
17	0.00000	0.00000	0.00000	-0.05010
18	0.00000	0.00000	0.00000	0.00000

```
DATAFILE NAME: WLS.DAT

   4 ITEMS: STARTING IN POSITION    1
COL  ITEM NAME          WDTH OPUT TYP N.DEC  ALTERNATE NAME
  1  LAY1                12   12  N    3
 13  LAY2                12   12  N    3
 25  LAY3                12   12  N    3
 37  LAY4                12   12  N    3


ENTER COMMAND >LI
```

$RECNO	LAY1	LAY2	LAY3	LAY4
1	252.708	261.011	266.257	268.977
2	265.738	267.447	270.568	272.719
3	267.484	273.175	276.816	279.227
4	999.000	280.730	284.364	287.119
5	999.000	282.369	290.284	288.370
6	999.000	999.000	283.795	285.603
7	999.000	999.000	284.066	285.764
8	999.000	999.000	999.000	288.232
9	999.000	999.000	999.000	290.884
10	252.353	260.799	266.249	269.086
11	262.013	266.718	270.697	273.215
12	267.201	274.573	278.190	281.668
13	999.000	288.885	288.243	298.152
14	999.000	283.246	286.530	289.310
15	999.000	999.000	283.271	285.614
16	999.000	999.000	283.957	285.734
17	999.000	999.000	999.000	288.227
18	999.000	999.000	999.000	290.883

Listing of INFO Program LOADWLS.PG

```
10000  PROGRAM SECTION ONE
10001 REM
10002 REM *Program: LOADWLS.PG
10003 REM *Run from WLSIN.CPL to calculate wls in .PAT file
10004 REM *equal to wls in newly get/copied data file WLS.DAT.
10005 REM *Also run from CBCIN.CPL to calculate flow terms
10006 REM *in .PAT file equal to newly get/copied data file
10007 REM *CBC.DAT.   Number of CA commands below must equal
10008 REM *number of model layers.
20000 PROG 2
20001 CA LAY1 = $1LAY1
20002 CA LAY2 = $1LAY2
20003 CA LAY3 = $1LAY3
20004 CA LAY4 = $1LAY4
30000 PROG 3
30001 END
```

Description of Fortran Program WLSIN.F77

A flag was necessary to tell the programs within the interface what model output was to be saved for return to ARC/INFO. One option available within the output-control package is to specify that output be saved (unformatted) on disk. The interface, using program WLSIN.F77, recognizes this as a flag. Any model output data set that has been saved on disk will be read from there, reformatted, and written to a file in a format acceptable for entry to INFO.

A default-naming convention was established for those files. The first three letters of the file name tell what parameter is saved. These are followed by a period '.' and a two-digit integer representing the stress period, followed by another period '.' and a two-digit integer representing the time step for which that parameter was saved. The three-character parameter codes are:

CODE	PARAMETER
HED	Head
DRD	Drawdowns
DRN	Flows to drain nodes
RIV	Flows to river nodes
WEL	Flows to well nodes
GHB	Flows to general head boundary nodes
EVT	Flows to evaporation nodes
RCH	Flows to recharge nodes
KHD	Flows to constant head nodes
STO	Storage changes for all nodes
CBR	Flow through the right face of all nodes
CBF	Flow through the front face of all nodes
CBL	Flow through the lower face of all nodes

Listing of Fortran Program WLSIN.F77

```
C-----------------------------------------------------------------------------
C     Program WLSIN.F77, run from WLSIN.CPL.  This program will read
C     the unformatted 'saved' files specified in the OC (output control)
C     file by variables IHEDUN and IDDNUN, and the requested cell-by-cell
C     flow files.  The program will then open and write one file
C     for each stress period for time step for each parameter saved
C     containing head, drawdown, and cell-by-cell flows to be 'GET/COPIED'
C     into INFO.
C
C
C     Written by Peter Van Metre
C     4/24/87; modified 10/23/89
C-----------------------------------------------------------------------------

C     ------------------------------------------
C--BUFHD must be dimensioned (nlay,nrow,ncol)
C--BUFFL must be dimensioned (ncol,nrow,nlay)
C--IOFLG dimensioned (nlay,4)

      dimension ioflg(4,4),bufhd(4,2,9),buffl(9,2,4),icb(10)
      character*9 name,fname(10)*3
      logical*4 lop,ex
C     ------------------------------------------

      data fname/'bcf','riv','rch','wel','drn','evt','ghb',3*' '/
      ierr=0
      open (unit=5,file='oc')
      inquire(file='wlsunf',exist=ex)
      if(ex)then
       open (unit=6,file='wlsunf',form='unformatted',status='old',
     .      iostat=ierr)
        if (ierr.ne.0) then
          print '(a)',
     .    'Error opening WLSUNF, bailing out of WLSIN.F77...'
          goto 901
        endif
      endif
      inquire(file='ddsunf',exist=ex)
      if(ex)then
       open (unit=30,file='ddsunf',form='unformatted',status='old',
     .      iostat=ierr)
        if (ierr.ne.0) then
          print '(a)',
     .    'Error opening DDSUNF, bailing out of WLSIN.F77...'
          goto 901
        endif
      endif
      open (unit=29,file='bas')
      do 3 j=1,7
       open (unit=31,file=fname(j),status='old',iostat=ierr)
       icb(j)=0
       if(ierr.eq.0)then
```

```
        if(fname(j).eq.'bcf' .or. fname(j).eq.'BCF')then
         read(31,101)iss,icb(j)
        else
         read(31,103)icb(j)
        endif
       junit=j+40
        if(icb(j).gt.0)then
         inquire(file=fname(j)//'unf',exist=ex)
         if(ex)then
          open (unit=junit,file=fname(j)//'unf',status='old',
     .          form='unformatted',iostat=ierr)
         endif
        endif
       endif
      close(31)
3     continue

      ibcfcb=icb(1)
      irivcb=icb(2)
      irchcb=icb(3)
      iwelcb=icb(4)
      idrncb=icb(5)
      ievtcb=icb(6)
      ighbcb=icb(7)
      data name /'hed.00.00'/

C--Read control information
      read(29,102)nlay,nrow,ncol,nper
      read(5,100)ihedun,iddnun

C--Check to see if head or drawdown saved for this simulation
      if(ihedun.lt.1 .and. iddnun.lt.1)then
       go to 901
      endif

4     continue
C--Read oc specifications for this time step
      read(5,101,end=901)incode,ihddfl,ibudfl,icbcfl

C--Decode incode to determine how to set flags
      if(incode)10,20,30
C--use ioflg from last step
10    continue
      go to 500
C--read ioflg for layer1 and assign to all layers
20    read(5,101) (ioflg(1,m),m=1,4)
      do 5 k=1,nlay
       ioflg(k,1)=ioflg(1,1)
       ioflg(k,2)=ioflg(1,2)
       ioflg(k,3)=ioflg(1,3)
       ioflg(k,4)=ioflg(1,4)
5     continue
      go to 500
C--read ioflg in entirety
```

```
30    read(5,101) ((ioflg(k,i),i=1,4),k=1,nlay)

C--Ioflg is set; read appropriate data sets and write
500   continue
      if(ihddfl.ne.0)then
C--heads and/or dd printed or saved
       iwrite=0
       iunit=0
        do 6 k=1,nlay
        do 6 i=1,nrow
        do 6 j=1,ncol
        bufhd(k,i,j)=0.0
6       continue
       do 15 k=1,nlay
        if(ioflg(k,3).ne.0)then          /*heads were saved
         read(6)kstp,kper,pertim,totim,text,nc,nr,ilay
         read(6)((bufhd(k,ir,ic),ic=1,ncol),ir=1,nrow)
111   format(2i5,4a4)
          iwrite=1
          tstp=kstp
          sper=kper
          sper=sper/100.
          tstp=tstp/100.
        endif
15    continue
       if(iwrite.gt.0)then
        write(name(1:3),'(a)')'hed'
        write(name(4:6),'(f3.2)')sper
        write(name(7:9),'(f3.2)')tstp
        do 7 io=50,139
         inquire(unit=io,opened=lop)
         if(.not. lop) then
          open(unit=io,file=name)
          iun=io
          go to 8
         endif
7       continue
8       continue
        do 16 ir=1,nrow
        do 16 ic=1,ncol
        write(iun,105)(bufhd(k,ir,ic),k=1,nlay)
16      continue
       endif

C--Now read and write drawdowns
       iwrite=0
        do 25 k=1,nlay
        do 25 i=1,nrow
        do 25 j=1,ncol
        bufhd(k,i,j)=0.0
25      continue
       do 35 k=1,nlay
        if(ioflg(k,4).ne.0)then              /*drawdowns were saved
         read(30)kstp,kper,pertim,totim,text,nc,nr,ilay
         read(30)((bufhd(k,ir,ic),ic=1,ncol),ir=1,nrow)
         iwrite=1
```

```
         sper=kper
         tstp=kstp
         sper=sper/100.
         tstp=tstp/100.
        endif
35     continue
       if(iwrite.gt.0)then
        write(name(1:3),'(a)')'drd'
        write(name(4:6),'(f3.2)')sper
        write(name(7:9),'(f3.2)')tstp
        do 17 io=31,139
         inquire(unit=io,opened=lop)
         if(.not. lop) then
          open(unit=io,file=name)
          iun=io
          go to 18
         endif
17      continue
18      continue
        do 36 ir=1,nrow
        do 36 ic=1,ncol
        write(iun,105)(bufhd(k,ir,ic),k=1,nlay)
36      continue
       endif
      endif

C--Check to see if cell-by-cell flow terms saved for this time
C--step.  If so call subroutine CBCIN to read those unformatted
C--files and write files for INFO to GET/COPY.

      if(icbcfl.ne.0)then
       if(ibcfcb.gt.0)then                    /* cell-by-cell terms saved
        if(iss.eq.0)then                      /* transient simulation
         write(name(1:3),'(a)')'sto'          /* storage
         ibdchn=41
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
         write(name(1:3),'(a)')'khd'          /* constant head nodes
         ibdchn=41
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        if(ncol.gt.1)then
         write(name(1:3),'(a)')'cbr'          /* right face, all nodes
         ibdchn=41
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(nrow.gt.1)then
         write(name(1:3),'(a)')'cbf'          /* front face, all nodes
         ibdchn=41
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(nlay.gt.1)then
         write(name(1:3),'(a)')'cbl'          /* lower face, all nodes
         ibdchn=41
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
```

```
        endif
       if(irivcb.gt.0)then
         write(name(1:3),'(a)')'riv'          /* river nodes
         ibdchn=42
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(irchcb.gt.0)then
         write(name(1:3),'(a)')'rch'          /* recharge nodes
         ibdchn=43
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(iwelcb.gt.0)then
         write(name(1:3),'(a)')'wel'          /* well nodes
         ibdchn=44
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(idrncb.gt.0)then
         write(name(1:3),'(a)')'drn'          /* drain nodes
         ibdchn=45
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(ievtcb.gt.0)then
         write(name(1:3),'(a)')'evt'          /* evapotranspiration nodes
         ibdchn=46
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
        if(ighbcb.gt.0)then
         write(name(1:3),'(a)')'ghb'          /* general head boundary nodes
         ibdchn=47
         call cbcin(nlay,nrow,ncol,name,ibdchn,buffl)
        endif
       endif
900    go to 4
901    continue
C--formats
100    format(t21,2i10)
101    format(4i10)
102    format(//,4i10)
103    format(t11,i10)
105    format(20f12.3)
       stop
       end


       subroutine cbcin(nlay,nrow,ncol,name,ibdchn,buffl)


C--Subroutine called from wlsin.f77 to read, reformat, and
C--write files in INFO/GET format for requested cell-by-cell
C--flow terms.


       dimension buffl(ncol,nrow,nlay),text(4)
       logical lop
       character name*9
```

```
C--Open output file, read unformatted records and write to output file.

      read(ibdchn)kstp,kper,text,nc,nr,nl
      read(ibdchn)buffl
        sper=kper
        tstp=kstp
        sper=sper/100.
        tstp=tstp/100.
        write(name(4:6),'(f3.2)')sper
        write(name(7:9),'(f3.2)')tstp
      write(1,101)(text(n),n=1,4),name
101   format(4a4,2x,a9)
      do 7 io=50,139
       inquire(unit=io,opened=lop)
       if(.not.lop)then
        open(unit=io,file=name)
        iun=io
        go to 8
       endif
7     continue
8     continue
      do 9 i=1,nrow
      do 9 j=1,ncol
      write(iun,100)(buffl(j,i,k),k=1,nlay)
9     continue

C--This format may need to be varied depending on units used and flows
C--generated.  If it is changed, item definitions in CBC.DAT INFO data
C--file must be changed to correspond.  For f12.5, info definition is:
C--item.name,12,12,n,5.
100   format(20f12.5)
      return
      end
```

APPLICATION OF THE INTERFACE PROGRAM

Developing a model in ARC/INFO using the interface can be accomplished by following a general series of steps. First the programs and INFO file definitions must be loaded to the users system. Copies of the interface have the two-column by nine-row by four-layer test model to illustrate the interface loaded to the INFO data files. In some cases, the steps listed below can be accomplished in several ways and in different order.

1. Generate a point coverage of the centers of model blocks with items row and column. In the test model, this coverage is called NODES.

2. Add item KEY to the NODES.PAT file using the ARC command ADDITEM.

3. Sort the NODES.PAT file on ROW, COLUMN and calculate item KEY = $RECNO. Sort back on NODES# before exiting.

4. Copy coverage NODES to create coverages BAS and BCF using ARC COPY. Add items for each nonconstant two-dimensional array to each coverages .PAT file. For example, if the model is three layers and starting heads are not constant, add items SHEAD1, SHEAD2, and SHEAD3 to BAS.PAT.

5. Load the appropriate data sets to each item. This can be accomplished in many ways and depends on how the original data are stored.

6. Enter controlling information to each package control file with the INFO ADD command, the INFO UPDATE command, or a PRIMOS level editor. The control file can be created at the PRIMOS level and brought into INFO using the "GET COPY" command. The .FMTS INFO data files associated with each package also may be created in the same manner.

7. Enter the array-control card information to each package .FMTS INFO data file using the INFO ADD command or the INFO UPDATE command if modifying an existing record.

8. Following the example files set up for each list formatted package load data for the packages.

9. Run program OUTALL or program OUTSEL from INFO and check the PRIMOS level ASCII files created. Either program can be run from INFO by typing the following sequence:

   ```
   Format $CHR39,4,C
   Move 'LIST' to $CHR39
   Run OUTALL
   ```
   Note: 'List' can be replaced by 'WIDE' and OUTALL can be replaced by OUTSEL.

10. Edit program WLSIN.F77 in the INFO directory to set the dimensions of BUFHD and BUFFL equal to the dimensions of your model. Recompile the program (F77 WLSIN) and reload the program (BIND -LO WLSIN -LI).

Three suggestions that may help avoid problems are:

1. Make sure the formats in the array-control cards correspond to the formats in the data files to which they relate. By default, INFO will insert one space between items output with the DISPLAY or LIST commands. This means that an item that has an INFO definition of 5,N,2 (5 spaces wide, numeric, 2 places to the right of the decimal point) will use six spaces when it is output to PRIMOS. If the item definition for SHEAD1, in BAS.PAT, therefore, is 5,N,2, the format for the SHEAD1 array-control card in BAS.FMTS should be F6.0.

2. INFO does not support 'E' format (scientific notation in Fortran). The only place 'E' format shows up in the example model is in item CONST in the .FMTS data files

(array-control cards). This item is defined as a character item, and values are entered as character strings. When the values are output to PRIMOS, these values are read as numbers in 'E' format by the model. There is a tradeoff by allowing for 'E' format in this way. An INFO calculate command cannot be used on these items. If this is a problem, redefine CONST as a numeric item and do not use 'E' format.

3. INFO program LOADWLS.PG must have one calculate statement for each model layer to properly load water levels or cell-by-cell flow terms to ARC/INFO coverages. For the example problem, LOADWLS.PG has four calculate statements. Also, when water levels or flow terms are loaded to INFO, they are temporarily loaded to either the WLS.DAT or the CBC.DAT INFO data files. Each of those data files must have one item for each model layer. Use ADDITEM or DROPITEM from ARC to give each of those files the correct number of items following the naming convention and formats specified (LAY#, 12, N, 3 for WLS.DAT and LAY#, 12, N, 5 for CBC.DAT.

SUMMARY

A computer-program interface between a geographic-information system and a ground-water flow model links two unrelated software systems for use in developing the flow models. The interface program allows the modeler to compile and manage geographic components of a ground-water model within the geographic-information system. A significant savings of time and effort is realized in developing, calibrating, and displaying the ground-water flow model.

Four major guidelines were followed in developing the interface program: (1) no changes to the ground-water flow model code were to be made, (2) a structure was to be designed within the GIS that follows the same basic structure as the ground-water flow model, (3) the interface program was to be flexible enough to support all options available within the model, and (4) the interface program was to be as efficient as possible in terms of computer time used and online-storage space needed.

The program was written in Fortran, control-program language and INFO program language. INFO data files and ARC/INFO coverages are defined to hold all model-related data. Output of data from INFO for ground-water model runs and return of model results to INFO are automated using control-program language programs. Because some programs are written in control-program language, the interface will work only on a PRIME computer using the PRIMOS operating system.

The interface program performs no data transformations or manipulations and is used only to conveniently store and move data between ARC/INFO and the ground-water flow model. The report documents an interface program that can be used to move data between a GIS and a ground-water flow model.

REFERENCES CITED

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.

_____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. (Supersedes U.S. Geological Survey Open-File Report 83-875)

Seybold, John, 1985, PRIME user's guide, 4th ed.: Natick, Massachusetts, PRIME Computer, Inc., DOC4130-4LA, Revision 19.4, v.p.

Environmental Systems Research Institute, 1987, ARC/INFO, Command references—The Geographic Information System Software: Redlands, California, Environmental Systems Research Institute, v. 2, variably paged.

GLOSSARY OF TERMS

Selected terms used in the text of this report are defined below. Additional definitions in the computer program listings of ground-water model variables can be found in McDonald and Harbaugh (1988), Seybold (1985), and Environmental Systems Research Institute (1987).

ADDITEM	An ARC/INFO command that will add an additional item to an INFO data file.
ARC/INFO	A commercially available GIS software system.
ARCPLOT	A plotting software package that is a subset of the ARC/INFO system.
ARRAY	An INFO item containing the name of the array that the array-control card corresponds to and must match the item name of that array in the corresponding .PAT file. The value stored in ARRAY is used to output its corresponding data values from the .PAT file.
ARRAY CONTROL CARD	A line in a model-input data file that contains information about the array of data to follow.
BAS	Basic package: Handles those tasks that are part of the model as a whole. Among those tasks are specification of boundaries, determination of time-step length, establishment of initial conditions, and printing of results.

BCF	Block-Centered Flow Package: Calculates terms of finite-difference equations that represent flow within porous medium; specifically, flow from cell to cell and flow into storage.
CPL	Command procedure language is a high-level programming language that operates at the PRIMOS command level (Seybold, 1985).
.CNTL	A control file that is an INFO data file containing information such as number of time steps, layers, rows, and columns used for output to the ground-water flow model.
Coverage	A digital map with associated information in ARC/INFO.
DELR	Cell width along the rows of the model.
DELC	Cell width along the columns of the model.
DRN	Drain Package: Adds terms representing flow to drains to the finite-difference equations.
EVT	Evapotranspiration Package: Adds terms representing evapotranspiration to the finite-difference equations.
.FMTS	INFO data files containing the array control cards for a particular ground-water flow model package.
FORTRAN	A language in which computer programs are written.
FMTIN	FORTRAN format statement in an array-control card describing the format of the array.
GHB	General-Head Boundary Package: Adds terms representing general-head boundaries to the finite-difference equations.
GIS	A geographic-information system that is a combination of digital mapping and display software system with a data-base management system used to manage, analyze, and display tabular and geographic data.
GRID	A polygon coverage of the model grid, used primarily for plotting purposes.
HNOFLO	Value assigned to head in inactive (no-flow) cells. It makes those cells stand out in listing of heads.
IBOUND	Array defining boundary conditions for a model layer.
IKEY	An integer item that equals 9 in all files and is used as part of the file-relating sequence in INFO that is used for all packages that produce two-dimensional arrays.
INEVTR	Maximum evapotranspiration rate read flag.
INEXDP	Evapotranspiration extinction depth read flag.

INFO — A computer software system designed for data-base and file management.

INIEVT — Evapotranspiration layer read flag.

INIT — An argument to an INFO command that opens PRIMOS files and initializes those files as they are opened. This procedure deletes existing values in those files.

INRECH — Recharge read flag.

INSURF — Evapotranspiration surface read flag.

ITEM — A variable defined in an INFO data file.

INIRCH — Recharge read flag.

IUNIT — A 24-element array that indicates which major options are to be used and the unit numbers from which input is to be read for the ground-water flow model.

.KEY — A .KEY file is an INFO data file containing record numbers for two other INFO data files and is used as an intermediary file when relating the two other files to each other. This results in much faster output of data and controls the order in which data are output.

KPER — An INFO item in a package .CNTL data file that contains the name of a different INFO data file that holds information to be output. For example, an array of values for the Evapotranspiration package to be output following an array-control card in EVT.CNTL file will be stored in an INFO data file whose name is specified in item KPER of the array-control card.

LOCAT — PRIMOS file unit location of the data that will be put in the array.

NODES — A point coverage of the center points of each model block and is the basic building block for each coverage that contains model data for all model blocks. For example, coverage, BAS, containing starting head values for each model block is a copy of coverage NODES.

NSTP — Number of time steps in a stress period.

NEVTOP — Evapotranspiration package option code.

PERLEN — Length of a model stress period.

.PAT	Point or polygon attribute file is the INFO data file created by ARC/INFO to hold information associated with a coverage. For example, if the coverage is a digital map of well locations called wells, each well will have a record in the INFO data file WELLS.PAT associated with that well location and will contain information about that well. Each piece of information, for example well depth, is an attribute.
PRIMOS	A mainframe computer and operating system.
RCH	Recharge Package: Adds terms representing areally distributed recharge to the finite-difference equations.
RIV	River Package: Adds terms representing flow to rivers to the finite-difference equations.
ROW	INFO item containing model row number that is necessary for arrays that are written across the page and are wider than 132 characters.
$RECNO	Record number for an INFO data file.
SEG command	PRIMOS command that will execute a FORTRAN program.
SIP	Strongly Implicit Procedure: Iteratively solves the system of finite-difference equations.
SOR	Slice-Successive Overrelaxation: Iteratively solves the system of finite-difference equations.
SORT	An INFO item containing an integer that is used to sort that INFO data file in a particular order.
SHEAD1	Starting head value or values for model layer 1.
TIN	A software package that is a subset of ARC/INFO that allows the storage and use of surface data (such as land surfaces) in a GIS.
TRPY	Ratio of transmissivity in the column direction to transmissivity in the row direction.
TSMULT	The multiplier for the length of successive time steps.
WEL	Well Package: Adds terms representing flow to wells to the finite-difference equations.