Documentation for *HYDMOD*, a Program for Extracting and Processing Time-Series Data from the U.S. Geological Survey's Modular Three-Dimensional Finite-Difference Ground-Water Flow Model

By R.T. Hanson *and* S. A. Leake

ABSTRACT

This report presents a FORTRAN computer program that generates simulated time-series data as output from the U.S. Geological Survey Modular Finite-Difference Ground-Water Flow Model at user-specified point locations or a collection of points that compose a profile through the modeled region. The program can save time-series data at user-specified locations for simulated water levels, drawdown, critical head, compaction, subsidence, streamflow, streamflow stage, and streamflow leakage. A set of locations also can be specified to create time-series profiles through the model. These data can be used to establish the performance of the model and can help with model calibration and other forms of flow-system and water-resource analysis.

INTRODUCTION

Current ground-water and surface-water model applications typically require site-specific model output data at multiple time intervals. The hydrograph program described in this report, called *HYDMOD*, allows the user to generate time-series data from the U.S. Geological Survey's Modular Finite-Difference Model (*MODFLOW*, McDonald and Harbaugh, 1988) for hydraulic head (water-level altitude), subsidence, and streamflow-related model output. This program is an efficient and compact method for saving user-specified time-series data from the simulation of ground-water and surface-water flow with *MODFLOW*. This approach reduces the amount of disk space and output required from the model needed to acquire detailed time-series data from a simulation, which in turn increases the efficiency of the model operation and reduces the time required for simulation. This approach also enhances the level of analysis that can be done to evaluate the performance of the simulation and compare simulation results with measured spatially and temporally varying observations.

Purpose and Scope

The purpose of this report is to describe *HYDMOD*, a program to save time-series information from user-specific locations from a simulation using the modular finite-difference ground-water flow model (*MODFLOW*) by McDonald and Harbaugh (1988). The time-series

data can be used to assess the performance of various model features included in a simulation of ground-water and surface-water flow. These data can be useful for model calibration or evaluation of proposed scenarios of water-resource development. The data can be saved from either steady-state or transient-state simulations and can represent time-series at individual points or sequences of contiguous points that represent profiles for individual times or time series. This report provides program documentation and a user's guide for *HYDMOD*. Data types that can be saved with *HYDMOD* are head, drawdown, critical head, compaction, subsidence, stream stage, streamflow into a cell, streamflow out of a cell, and the flow of water between a stream and the underlying aquifer. *HYDMOD* is compatible with all other *MODFLOW* packages and with upgrades presented in *MODFLOW*-96 (Harbaugh and McDonald, 1996a). The *HYDMOD* program uses the additional utility subroutine URWORD that is included in *MODFLOW*-96 (Harbaugh and McDonald, 1996b). This additional subroutine must be added to earlier versions of *MODFLOW* to use *HYDMOD*. *HYDMOD* is modular and retrieves data separately from each *MODFLOW* package. Therefore, if time-series model data from additional features from existing packages or from future packages are required, the user can add additional new *HYDMOD* subroutines or modify existing subroutines as needed.

IMPLEMENTATION OF THE HYDROGRAPH PROGRAM (*HYDMOD*)

Four steps are required to create time-series data from *MODFLOW* using *HYDMOD* (fig. 1). First, the user creates an input data set that specifies the locations and types of output data desired from a simulation. Second, *MODFLOW* is run with *HYDMOD* to create a binary file of the specified data at the specified locations. Third, the binary file is processed with the post-processor program HYDFMT into an ASCII-text data file. And, fourth, the data can be plotted directly or entered into a spreadsheet. As a quick alternative to this approach, the program HYDPOST is also described in this report. HYDPOST can extract time-series data from certain unformatted (binary) output files generated by *MODFLOW* without the use of *HYDMOD* (Appendix 3).

Approach

The *HYDMOD* program consists of five subroutines (modules) that are called from the main program of *MODFLOW*. The program includes two additional subroutines (submodules) that are called by the *HYDMOD* program. At the start of a simulation, the *HYDMOD* program reads geographic coordinates for a specified number of observation points that are requested for the basic, streamflow-routing, and interbed-storage packages. Observation points need not correspond to cell centers in the model grid. The coordinates, XL and YL, are referenced to the lower-left corner of the model grid (fig. 2). The program also reads an index that corresponds to a desired output type, such as head, and the number of the model layer for which the output is desired. Before the simulation begins, the *HYDMOD* program locates the four surrounding model-cell centers (nodes). For example, the observation point OB-1 is surrounded by cell centers located at Z_1, Z_2, Z_3, and Z_4 (fig. 2). The user has the option of saving the value of the user-specified item from the cell center that coincides with the location of the observation point. For example, the point OB-1 occurs in the cell from the second row and third column (2,3) and the cell value from the cell center Z_2 would be used to represent the value for point OB-1 (fig. 2) if the cell value is requested. The user also has the option of estimating the value at the

actual location of the point. If the value at the actual location is desired, *HYDMOD* uses bilinear interpolation to estimate the value from the values of the surrounding cell centers. The bilinear-interpolation weights w_1, w_2, w_3, and w_4 are computed so that the interpolated value, $\hat{z}$, can be calculated as:

$$\hat{z} = w_1 Z_1 + w_2 Z_2 + w_3 Z_3 + w_4 Z_4,$$

where Z_1, Z_2, Z_3, and Z_4 are values of the simulated variable at the surrounding cell centers. The weights sum to 1 and are proportional to the relative distances between the specified location and the four surrounding cell-center locations. For each observation point, the four weights are stored so that the interpolation can be carried out for each time step. Four locations of surrounding cells are stored as integer pointers to elements in the X array of *MODFLOW*. For example, the observation point OB-1 would be interpolated from the surrounding cell centers, Z_1, Z_2, Z_3, and Z_4 (fig. 2).

When interpolation is not appropriate, such as near the boundary of the active part of the ground-water flow model or across a horizontal-flow barrier within the flow model, the user has the option to specify output of the data from the cell in which the observation point is located. A label header is also output that identifies the total number of observation points and lists a label that identifies each column of output data; this also helps maintain some level of flexibility for the order of observation points within the input file. Because potentially different topologies can be used to construct streamflow networks (Prudic, 1989, fig. 1), the segment and reach numbers of the stream reach in the streamflow-input data set are used as the reference coordinates for identifying stream reaches for saved output. The hydrograph program binary output also stores the user-specified time units used for simulation to facilitate correct time conversion by the post-processor program HYDFMT.

At the start of the simulation, the *HYDMOD* program writes a single record that includes the number of observation points to be recorded in each time step and the time units used for the simulation. For each time step, the *HYDMOD* program writes an unformatted record that includes total simulation time and a value for the user-specified item from the user-specified location. These unformatted records are written to an output unit specified by the user. After the simulation is complete, the unformatted output records can be read into another program to extract columns of data in ASCII-text form for plotting or tabulating. If a well is located in or adjacent to a cell that is an inactive cell or a cell that has gone dry, the interpolated value is set to the user-specified HYDNOH value. A *HYDMOD* hydrograph post-processor program, HYDFMT, which reads the unformatted data and writes columns of formatted data in ASCII-text form, is given in Appendix 2.

Changes to the *MODFLOW* Program Required to Run *HYDMOD*

Six calls to the five subprograms of the *HYDMOD* program must be added to the Main program of *MODFLOW*. The locations of calls are referenced to numbers of comment statements in the version of the Main program dated December 3, 1996 (Harbaugh and McDonald, 1996a; McDonald and Harbaugh, 1988). In the **MAIN** program add the following:

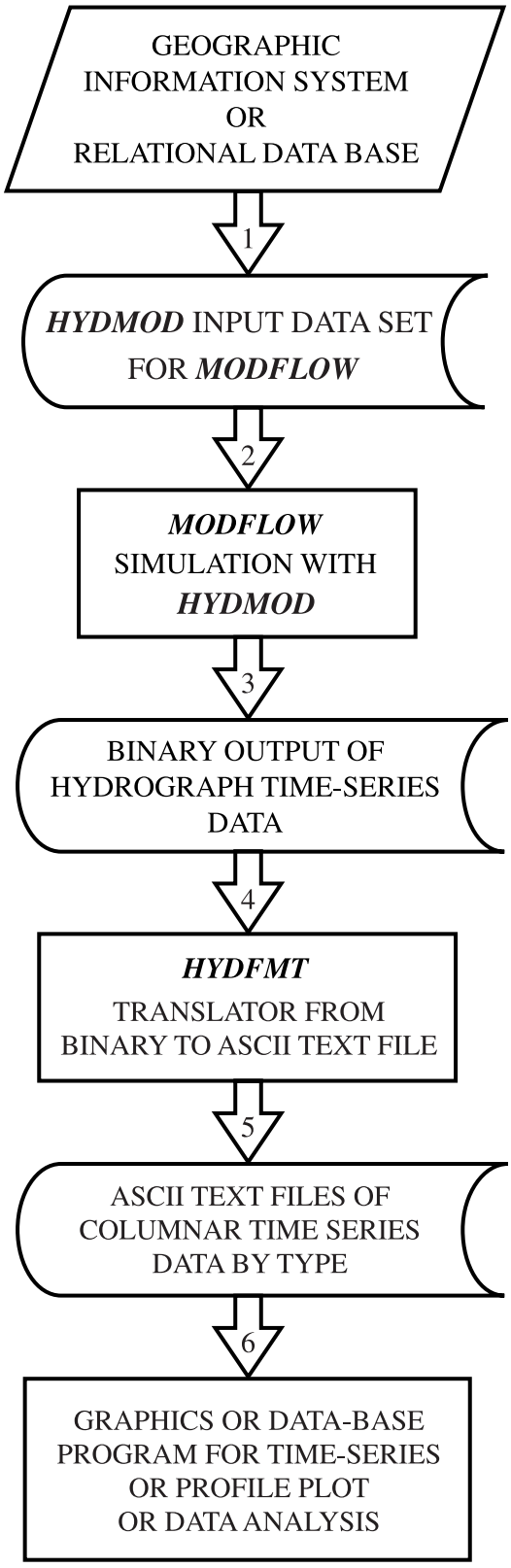
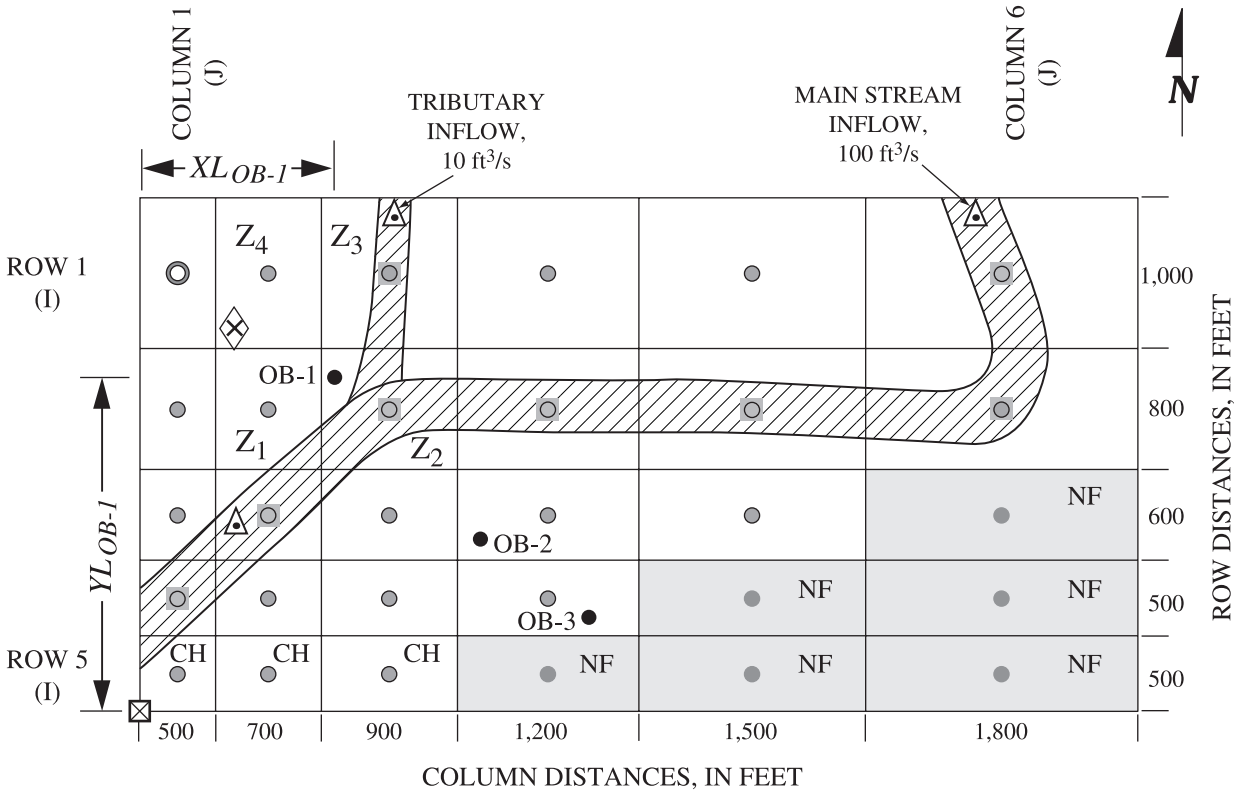


Figure 1. Flowchart showing the flow of time-series data for the *HYDMOD* program.



EXPLANATION

- Pumped-well location
- Observation-well location
- Extensometer-well location
- Bench-mark location
- Streamflow-gaging station location
- Model cell center (or node)
- Model cell with river
- Geographic local origin of model-related features such as well and bench-mark locations

CH — Model cell with user-specified fixed head (constant-head) set to 370 feet above sea level

NF — Model cell inactive in simulation (no-flow cell)

(XL_{OB-1}, YL_{OB-1}) — Example of geographic (x,y) coordinates of user-specified feature (OB-1) in local-coordinate system with origin in lower-left corner of model grid

Z_1, Z_2, Z_3, Z_4 — Value of the simulated variable at cell centers surrounding location of user-specified comparison point (OB-1)

Figure 2. Diagram showing test problem layout and coordinates used for interpolation.

Prior to comment C5, add the following two lines:

```
IF(IUNIT(22).GT.0) CALL HYDMAL(ISUM,LENX,LCHYDM,LCIRR,NHYDM,
     1           IHYDMUN,HYDNOH,IUNIT(22),IOUT)
```

Prior to comment C7, add the following seven lines:

```
      IF(IUNIT(22).GT.0) CALL HBASRP(X(LCHYDM),X(LCHOLD),NHYDM,NUMH,
    1        X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,LCHNEW,
    2        LCIBOU,IUNIT(22),IOUT)
     IF(IUNIT(22).GT.0.AND.IUNIT(19).GT.0)
    1        CALL HIBSRP(X(LCHYDM),NHYDM,NUMH,
    2        X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,
    3        LCIBOU,LCSUB,LCHC,IUNIT(21),IOUT)
```

Prior to comment C7C, add the following eight lines:

```
     IF(IUNIT(22).GT.0.AND.IUNIT(18).GT.0.AND.KPER.EQ.1)
    1        CALL HSTRRP(X,ISUM,X(LCHYDM),NHYDM,NUMH,
    2        X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,
    3        LCIBOU,LCSTRM,ICSTRM,NSTREM,IUNIT(22),IOUT)
C
C ------WRITE STARTING HYDROGRAPH RECORD
      IF(IUNIT(22).GT.0.AND.KPER.EQ.1) CALL HYDMOT(X,ISUM,X(LCHYDM),
    1        NUMH,IHYDMUN,0.0,HYDNOH,NROW,NCOL,ITMUNI,IOUT)
```

Prior to comment C7C5, add the following four lines:

```
C
C ------WRITE HYDROGRAPH RECORD
    IF(IUNIT(22).GT.0) CALL HYDMOT(X,ISUM,X(LCHYDM),NUMH,
   1          IHYDMUN,TOTIM,HYDNOH,NROW,NCOL,ITMUNI)
```

APPLICABILITY AND LIMITATIONS

The *HYDMOD* program reduces the need for large output files and facilitates the analyses of hydraulic data from *MODFLOW*. Data from *HYDMOD* can be used to construct water-level hydrographs from wells, bench-mark trajectories resulting from compaction and subsidence, and streamflow hydrographs from gaging stations or diversions. *HYDMOD* can be used to create stream profiles of gaining and losing stream reaches for steady- or transient-state simulations. *HYDMOD* can be used to create profiles of drawdown, water levels, compaction, subsidence, and critical head. *HYDMOD* can also be used to extract time-series of river stage, preconsolidation critical head that controls subsidence, or drawdown from initial water-level conditions. *HYDMOD* allows the user to specify a flag for data that occur at sites outside the active flow region or that occur in cells that become permanently or temporarily dry and inactive. *HYDMOD* can process as many points as the user specifies and is limited only by the size of the X vector of *MODFLOW* (McDonald and Harbaugh, 1988, Appendix B-1). The post-processor program HYDFMT is set to process as many as 5,000 individual locations for a specific simulation, but this can be enlarged to any desired number by simply increasing array dimensions.

Although *HYDMOD* also allows the user to interpolate a value at the specified location from the four surrounding cells, the use of bilinear interpolation assumes that the attribute varies linearly within the region of the four surrounding cells. A user-specified location may occur where there is a large transmissivity difference between cells, adjacent to a fault represented with the horizontal-flow barrier package, adjacent to the edge of the model, or adjacent to an inactive region within the model. For these types of locations bilinear interpolation may not be appropriate, and the nearest cell option for retrieving the value of the attribute should be used. Interpolation occurs only within the layer and not between layers. Interpolation also is not available for any of the streamflow attributes because the topology of the stream network and actual river miles are variable from one reach to another. If needed, additional interpolators could be added to the *HYDMOD* program.

HYDMOD PROGRAM INPUT INSTRUCTIONS

Input to the *HYDMOD* program is read from the file name and unit specified in name file for *MODFLOW*-96. The file type that is added to the name file for *HYDMOD* is specified as HYD. The name 'HYD' is also added to the twenty-second element of the character array, CUNIT, that is specified in the data statement in the specifications of the **MAIN** program of *MODFLOW*-96.

FOR EACH SIMULATION

HYDMAL

1. Data: NHYD IHYDUN HYDNOH

Format [1]: I10 I10 F10.0

HYDMRP

2. Data: PCKG ARR INTYP KLAY XL YL HYDLBL

Format: A3 A2 A1 I3 F10.0 F10.0 A20

(Input item 2 consists of one record for each hydrograph. No more than NHYD hydrograph records will be read.)

Explanation of Fields Used in Input Instructions

NHYD is the maximum number of observation points that may be read in.

IHYDUN is a unit number to which hydrograph records are written.

HYDNOH is a user-specified numeric index that is output for wells that are either located outside the interpolateable active flow region, located in a dry cell, or located in a cell that may rewet but is temporarily dry.

PCKG is a flag to indicate which package and which array in the X vector is to be addressed by *HYDMOD* for the hydrograph of each observation point. As new model packages are added, the Read-and-Prepare module can be appended with additional subroutines to access new packages or new arrays. These are the same names specified in the CUNIT array. The three current package options are the basic (BAS), streamflow routing (STR), and interbed storage (IBS) packages.

ARR is an index to indicate which model attribute is to be accessed for the hydrograph of each observation point. The table of arrays available for access is as follows:

[1]Although fixed formats are shown here, *MODFLOW*-96 accepts free format with each variable delineated by a comma or a blank.

ARR Value	Feature to be recorded by *HYDMOD*
HD	Head array (HNEW, BAS Package)
DD	Drawdown array (HOLD-HNEW, BAS Package)
HC	Preconsolidation-head array (HC, IBS1 Package)
CP	Compaction array (SUB, IBS1 Package)
SB	Subsidence array (SUB, IBS1 Package)
ST	Streamflow-stage array (STR Package)
SI	Streamflow into reach array (STR Package)
SO	Streamflow out of reach array (STR Package)
SA	Streamflow into aquifer array (STR Package)

INTYP is an index to indicate how the data from the specified feature are to be accessed. The current two options are the interpolated value (I) or the cell value (C) for the entire hydrograph of each observation point.

KLAY is the layer sequence number of the array to be addressed by *HYDMOD*. For arrays that exist for each model layer, such as head (HNEW), KLAY is the layer number. For arrays that exist for only certain model layers, KLAY is the sequence number of the array starting with the uppermost array as 1, the next lower array as 2, and so forth. For the subsidence option (ARR = 'SB'), KLAY designates the range of layers (from layer 1 to layer "KLAY") used to sum compaction from individual layers into total subsidence for these layers.

XL is the coordinate of the hydrograph point in model units of length measured parallel to model rows, with the origin at the lower left corner of the model grid (fig. 2). The value of XL does not need to correspond with the location of any particular model node. For streamflow-related data, XL designates the segment number that contains the reach from the streamflow-input data set for which data are to be output.

YL is the coordinate of the hydrograph point in model units of length measured parallel to model columns, with the origin at the lower left corner of the model grid (fig. 2). The value of YL does not need to correspond with the location of any particular model node. For streamflow-related data, YL designates the reach number that is the cell location from the streamflow-input data set for which data are to be output.

HYDLBL is the label of the hydrograph point in the model and is a total of 20 characters in length. The purpose of the label is to identify output and allow users to reorder, add, or subtract from the input and still maintain location identity. The location label of any particular hydrograph point has the model attribute, ARR (characters 1 and 2), the INTYP value (character location 3), and KLAY value (character locations 4 to 6) written as a prefix into the first six characters of the label and saved to the unformatted output file. The remainder of the original label is shifted to the right, yielding a net 14 characters available for location label to final output from the input label. For streamflow, the model attribute, stream-segment number (character locations 3 to 5), and stream-reach number (character locations 6 to 8) are written into the beginning of the label, yielding a net 12 characters available for location label to final output from the input label. The records containing the character variable, HYDLBL, should end in a blank—otherwise, the last character will be truncated by the utility-reading subroutine URWORD (Harbaugh and McDonald, 1996b).

TEST PROBLEM

A simple test problem is used to demonstrate how to set up the input data and give some selected examples of the types of model data that can be generated with *HYDMOD*. A simulation was run with this simple problem to demonstrate the application of selected features of the *HYDMOD* program. The test problem shows how time-series data from the simulation can be generated for:

(1) Drawdown at a pumped well;

(2) Multiple hydrographs from multiple layers that could represent multiple-well observation sites at different distances from the pumped well;

(3) Compaction from a specific layer that could represent data from an extensometer well;

(4) Subsidence as the sum of compaction from all layers with simulated compaction that could represent an extensometer well or bench mark;

(5) Streamflow out of a cell that could represent flow past a streamflow-gaging station; and

(6) Flow between a stream and aquifer in a cell that could represent streamflow infiltration to the upper aquifer layer or ground-water discharge to the stream.

The test problem is simulated with a finite-difference grid consisting of three layers that represent three aquifers. All layers contain five rows and six columns of cells (figs. 2, 3). The first three cells in row 5 of layer 1 (the top layer) are a constant-head boundary. The dimensions of the cells are variable in both the row and column directions (fig. 2). The upper layer has a transmissivity of 2,000 ft^2/d, and the middle and lower layers have a transmissivity of 1,000 ft^2/d. The upper layer has an unconfined (specific-yield type) storage coefficient of 0.1, and the middle and lower layers have confined storage coefficients of 0.00005 and 0.00001, respectively. The vertical conductance between the upper and middle layers and between the middle and lower layers is 0.0001 d^{-1} and 0.00001 d^{-1}, respectively. The initial conditions include a constant uniform recharge of 0.008 ft/d. There is a stream that flows into the model and joins with a smaller tributary to form one outflow stream. The stream inflow was set to 100 ft^3/s, and the tributary inflow was set to 10 ft^3/s. All three layers are subject to subsidence, have no prior subsidence, and are assumed to be initially in a state of normal consolidation. Thus, the preconsolidation (critical) heads are set to the steady-state solution of heads for all cells in all layers. The elastic storage coefficients are set to 0.0001 and the inelastic storage coefficients are set to 0.001 for all cells in all three layers. For the steady-state simulation there are no pumped wells, and for the transient simulation there are two wells located in cell (1,1) that are pumped at about 208 gallons per minute (gal/min) from the top aquifer (layer 1) and at about 104 gal/min from the middle aquifer (layer 2). The duration of the transient simulation is 2.74 years, which is distributed over 20 time steps that increase in length by a factor of 1.2. Data sets for the test problem, including input and selected output for the transient-state problem, are given in Appendix 1.

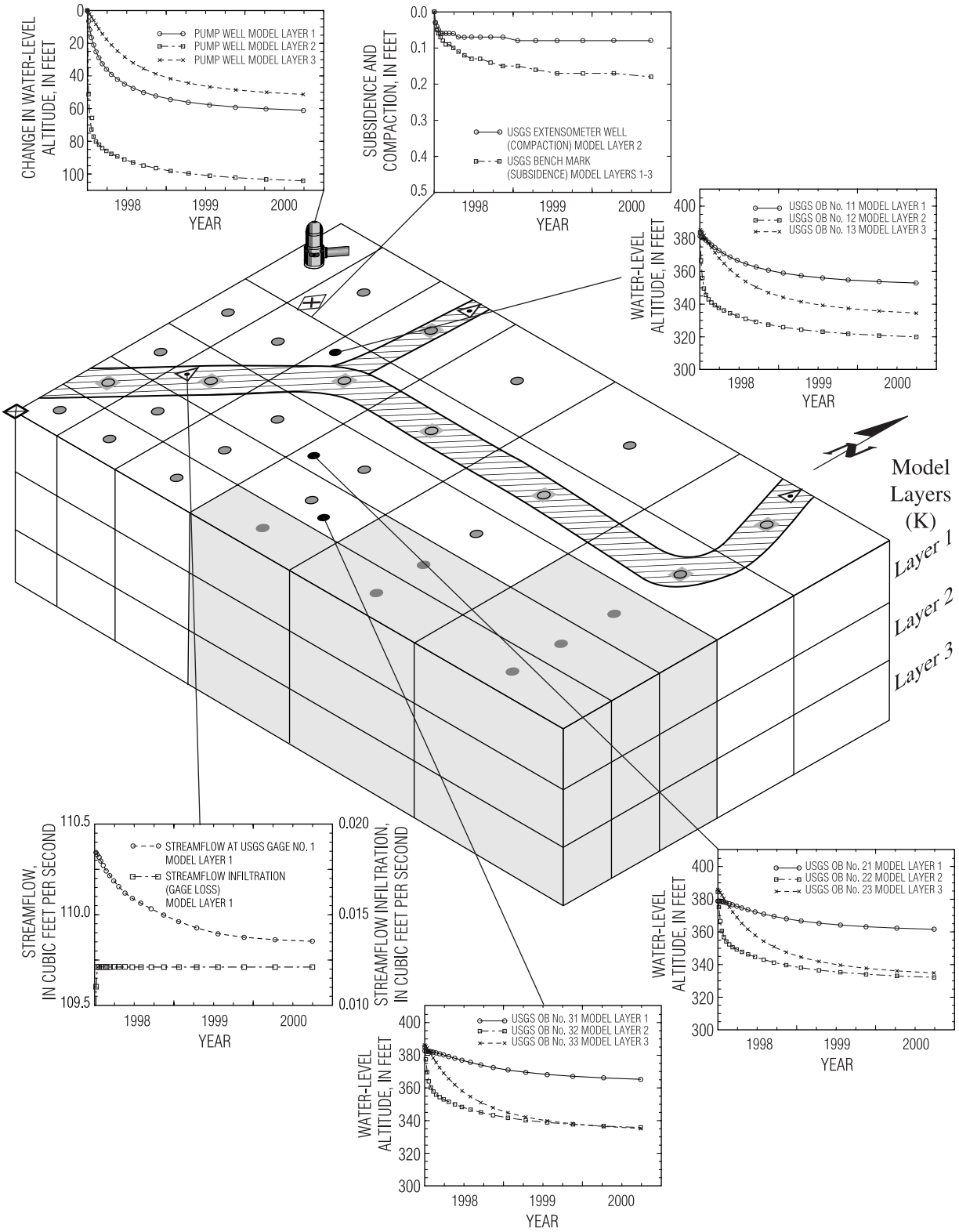


Figure 3. Isometric diagram showing test problem layout and results. (For explanation of map symbols see figure 2.)

To generate the input data set for using *HYDMOD* with the test problem, the locations and types of data were selected to be output. Then the locations of the observation points, except those that are stream related, were determined relative to the lower-left-hand corner of the model grid, which represents the model-grid-based geographic (x,y)-local origin for the locations of comparison points. The y-direction is aligned with the model rows and increases in distance from the local origin with decreasing row number. The x-direction is aligned with the model columns and increases in distance from the local origin with increasing column number. This model grid is aligned with rows parallel to the east–west direction and columns aligned with the north–south direction. If the site locations are referenced with global geographic coordinates such as latitude–longitude, universal transverse mercator, or local public-survey coordinates, then geographic or translation transformations may be used to convert the global coordinates into the model local geographic coordinates. If the model grid was rotated, then a simple set of rotation transformations also may be required to transform global geographic coordinates into the model local geographic coordinates. Most geographic information systems contain transformations or tools for building transformations to a local coordinate system. The assemblage of the observation points and transformation to local coordinates represents step 1 in using the *HYDMOD* program (fig. 1).

The input data set for *HYDMOD* begins with a header record that contains the total number of observation points, the FORTRAN unit number from which this data set will be read by *MODFLOW,* and a value for HYDNOH. For this test problem there are 16 comparison points, the *HYDMOD* input data set will be read from unit 33, and a value of 998.0 was used for HYDNOH. The header record for the *HYDMOD* input data file is as follows:

```
16          33        998.0
```

Note that in this example the value selected for HYDNOH is different from the value used for HNOFLO, which is specified in the Basic Package input data for *MODFLOW.*

To retrieve drawdown data from the location of the pumped well, which is located in the corner of the model grid in cell (1,1), select an ARR value of DD. Because this is a corner cell where interpolation is not possible, a cell value is retrieved by using the INTYP value of C. This is done to retrieve drawdown data for all three aquifers (layers), thus yielding three comparison points that include the (x,y)-well location and a label as follows (Appendix 1):

PCKG	ARR	INTYP	LAYER(K)	XL	YL	LABEL
BAS	DD	C	1	250.0	2900.	PUMP_WELL
BAS	DD	C	2	250.0	2900.	PUMP_WELL
BAS	DD	C	3	250.0	2900.	PUMP_WELL

To retrieve water-level altitudes (heads) from the three multiple-well observation sites that extend outward radially from the pumped well, a PCKG value of BAS and an ARR value of HD are specified. The observation wells located in cells (2,3) and (3,4) will have the water level interpolated to the well location on the basis of water levels simulated at the four surrounding active cells. The farthest observation well located near the no-flow boundary in cell (4,4) cannot be interpolated. For this comparison point an INTYP value of C is used to retrieve the cell value of water-level altitude. At all three sites water-level-altitude data are retrieved for all three layers, yielding nine additional records with labels as follows (Appendix 1):

PCKG	ARR	INTYP	LAYER(K)	XL	YL	LABEL
BAS	HD	I	1	1400.0	2200.	USGS_OB_No_11
BAS	HD	I	2	1400.0	2200.	USGS_OB_No_12
BAS	HD	I	3	1400.0	2200.	USGS_OB_No_13
BAS	HD	I	1	2200.0	1200.	USGS_OB_No_21
BAS	HD	I	2	2200.0	1200.	USGS_OB_No_22
BAS	HD	I	3	2200.0	1200.	USGS_OB_No_23
BAS	HD	C	1	3000.0	700.	USGS_OB_No_31
BAS	HD	C	2	3000.0	700.	USGS_OB_No_32
BAS	HD	C	3	3000.0	700.	USGS_OB_No_33

To retrieve compaction from a specific layer that could represent data from an extensometer well located near the pumped well, select the PCKG value of IBS, an ARR value of CP, and an INTYP value of I that will yield an interpolated value of compaction for the middle aquifer (layer 2) since KLAYER is 2. Simulated subsidence can represent the sum of compaction from a subset or all layers with compaction active. In this example an additional comparison can be made between layer-specific compaction (for example, compaction only from layer 2) and total compaction (subsidence) by specifying the lowest active compacting layer (layer 3) to output total compaction of layers 1 through 3. The subsidence could represent an additional extensometer well completed in the lowest layer or a bench mark located on the land surface at the extensometer-well site. To retrieve a single-cell value of subsidence at this same location, select the PCKG value of IBS, an ARR value of SB, and an INTYP value of I. These two additional records also include the (x,y)-well location and a label as follows (Appendix 1):

PCKG	ARR	INTYP	LAYER(K)	XL	YL	LABEL
IBS	CP	I	2	600.0	2500.	USGS_EXTENSOMETER
IBS	SB	I	3	600.0	2500.	USGS_BENCHMARK

To retrieve streamflow flowing out of a cell that could represent flow past a streamflow-gaging station, the PCKG value of STR, an ARR value of SO, and an INTYP value of C are specified. To retrieve flow between stream and ground water in a cell that could represent streamflow infiltration to the upper aquifer layer, the PCKG value of STR, an ARR value of SA, and an INTYP value of C are specified. For retrieving any streamflow-related data with *HYDMOD*, the segment and reach numbers are specified in place of geographic coordinates for XL and YL, respectively. These two additional records also include a label as follows (Appendix 1):

PCKG	ARR	INTYP	LAYER(K)	XL	YL	LABEL
STR	SO	C	1	3.	2.	USGS_STREAM_GAGE
STR	SA	C	1	3.	2.	STREAM_LOSS_GAGE

Thus a total of 16 records that represent 3 drawdown time-series at the pumped well, 9 water-level-altitude time-series at the observation wells, 2 subsidence time-series at the extensometer well, and 2 streamflow time-series at the streamflow-gaging station compose the input data set that represent step 2 (fig. 1) of using the *HYDMOD* program with *MODFLOW*. The simulation is run with *MODFLOW*, generating one file of binary output (step 3, fig. 1). Then the binary output is translated into multiple ASCII-text files of columnar time-series data using the post-processor program HYDFMT (steps 3, 4, fig. 1). And finally, the six separate output files of water-level altitudes, drawdown, compaction, subsidence, streamflow out of the reach (cell), and ground-water/surface-water flow are graphed to assess the results of the test-problem simulation (steps 5, 6, fig. 1).

The results of any simulation can be viewed spatially with a sequence of locations that show profiles for all model time steps and can be viewed temporally at any single location for all model time steps with output from the *HYDMOD* program. For this test problem, the simulated data from 16 observation locations were retrieved to demonstrate some of the types of data that can be used to assess the performance of the simulation (fig. 2) at individual locations. Within the 2.74 years of pumping from cell (1,1,1), water-level decline (drawdown) was retrieved from all three layers at the pumped well and ranged from about 50 to 103 ft (fig. 3), with the largest drawdowns occurring in the middle layer. The three multiple-well observation sites (USGS_OBWELL_No_11 - 13, USGS_OBWELL_No_21 - 23, and USGS_OBWELL_No_31 - 33) extend radially out from the pumped well, across the river, and to the no-flow boundary in the southeastern part of the model (fig. 2). The interpolated water-level altitudes at the two closer multiple-well observation sites and the cell-value water-level altitudes at the farthest site (adjacent to the no-flow boundary) indicate different degrees of decline and changes in the relative position of water-level altitudes for the three layers at different distances from the pumped well. Compaction from an extensometer completed in the middle aquifer (layer 2) and total compaction representing subsidence at a bench mark on the top of the upper aquifer (layer 1) are located radially out between the pumped well and the closest multiple-well observation site (fig. 2). These interpolated data indicate that about 44 percent of the total subsidence is occurring in the middle layer where about 41 percent of the pumping is occurring (fig. 3). The streamflow at the downstream gaging station and the loss of streamflow into the upper aquifer (layer 1) were also retrieved (fig. 2). These data indicate that most of the streamflow passes through the system with a total loss of streamflow of about 0.49 ft^3/s. The streamflow loss was small (less than 0.03 ft^3/s) and a relatively constant amount infiltrates into the ground-water system in the one cell where the downstream streamflow-gaging station is located (fig. 3). Profiles of streamflow or streamflow loss could also have been generated by retrieving the various streamflow features from a sequence of cells in continuous downstream order. Similarly, distance-drawdown profiles also could have been generated by retrieving the drawdown radially outward from the pumped well from a sequence of cells or well sites in continuous outward order. These types of point and profile time-series data greatly enhance the ability of the modeler to assess the performance of the simulation and the changes in potentially complex suites of hydrologic components that operate in concert to yield results through time that cannot be ascertained from spatial analysis alone.

PROGRAM LISTING

Allocate Subroutine

```
      SUBROUTINE HYDMAL(ISUM,LENX,LCHYDM,LCIRR,NHYDM,IHYDMUN,HYDNOH,
     1IN,IOUT)
C
C-----VERSION 0100 29JULY1998 HYDMAL
C     ***************************************************************
C     ALLOCATE ARRAY STORAGE FOR HYDROGRAPH PROGRAM
C     ***************************************************************
C
C       SPECIFICATIONS:
C     -------------------------------------------------------------
      CHARACTER*80 LINE
C     -------------------------------------------------------------
C
C1------IDENTIFY PROGRAM.
      WRITE(IOUT,1) IN
    1 FORMAT(1H0,'HYDM -- HYDROGRAPH PROGRAM, VERSION 1.0,',
     1     ' 07/29/98',' INPUT READ FROM UNIT',I3)
C
C4------READ NUMBER OF HYDROGRAPHS AND UNIT FOR SAVING UNFORMATTED
C4------HYDROGRAPH FILE AND NUMERIC FLAG FOR DRY/INACTIVE CELLS
      READ(IN,'(A)') LINE
      LLOC=1
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,2,NHYDM,R,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,2,IHYDMUN,R,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,HYDNOH,IOUT,IN)
      WRITE(IOUT,5) NHYDM,IHYDMUN,HYDNOH
    5 FORMAT(1X,'      NUMBER OF HYDROGRAPH POINTS:',I4,
     1     /,1X,'HYDROGRAPHS WILL BE SAVED ON UNIT:',I4,
     2     /,1X,'PERIODS FOR DRY/INACTIVE CELLS WILL USE',
     3        ' THE NUMERIC FLAG:',F10.2)
C
C3------SET LCHYDM EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCHYDM=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY THE HYDROGRAPH LIST.
      ISUM=ISUM+18*NHYDM
      LCIRR=ISUM
      ISUM=ISUM+NHYDM
      ISP=18*NHYDM
      ISUM=ISUM+ISP
      ISP=18*NHYDM
C
C5------PRINT AMOUNT OF SPACE USED BY THE HYD PROGRAM
      WRITE(IOUT,6) ISP
    6 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR HYDROGRAPHS')
      ISUM1=ISUM-1
      WRITE(IOUT,7) ISUM1,LENX
    7 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,8)
    8 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6------RETURN
      RETURN
      END
```

Read-and-Prepare Subroutine

```
      SUBROUTINE HBASRP(HYDM,HOLD,NHYDM,NUMH,IHYDMUN,DELR,DELC,
     1NCOL,NROW,NLAY,LCHNEW,LCIBOU,IN,IOUT)
C
C
C-----VERSION 0100 29JULY1998 HBASRP
C     ******************************************************************
C     READ BASIC PACKAGE DATA FOR HYDROGRAPHS
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER HYDLBL*20,PTSLBL(5000)*20,LINE*80
      CHARACTER PCKG*3,ARR*2,INTYP*1
      DIMENSION HYDM(18,NHYDM),DELR(NCOL),DELC(NROW)
      DIMENSION HOLD(NCOL,NROW,NLAY)
      LOGICAL IBCHK
      COMMON /HYDCOM/ PTSLBL
C     ------------------------------------------------------------------
      NIJ=NROW*NCOL
      NCOL1=NCOL-1
      NUMHP=0
      REWIND(IN)
      READ(IN,'(A)',END=99) LINE
 10   READ(IN,'(A)',END=99) LINE
      LLOC=1
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      IF(LINE(ISTART:ISTOP).NE.'BAS') GO TO 10
      PCKG=LINE(ISTART:ISTOP)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      ARR=LINE(ISTART:ISTOP)
      WRITE(HYDLBL(1:2),FMT='(A2)')ARR
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      INTYP=LINE(ISTART:ISTOP)
      WRITE(HYDLBL(3:3),FMT='(A1)')LINE(ISTART:ISTOP)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,2,KLAY,R,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,XL,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,YL,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,0,N,R,IOUT,IN)
      WRITE(HYDLBL(4:6),FMT='(I3.3)')KLAY
      HYDLBL(7:20)=LINE(ISTART:ISTART+14)
cc    TIME SERIES FROM THE HEAD ARRAY
      IF (ARR.EQ.'HD') THEN
        LOC=LCHNEW
        NW=2
        ITYP=1
        IBCHK=.TRUE.
cc    TIME SERIES FOR THE DRAWDOWN ARRAY
      ELSE IF (ARR.EQ.'DD') THEN
        LOC=LCHNEW
        NW=2
        ITYP=2
        IBCHK=.TRUE.
      ELSE
        WRITE(IOUT,25) LINE
 25     FORMAT(' Invalid array type was found on the following',
     &   ' record:',/,A80)
        WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
```

```
        GO TO 10
      ENDIF
      CALL GRDLOC(XL,YL,DELR,DELC,NROW,NCOL,NR1,NC1,NR2,NC2,
     & X1,X2,Y1,Y2)
      IF(INTYP.EQ.'C') THEN
        NWT=1
        IF(NR1.LT.1.OR.NR1.GT.NROW.OR.NC1.LT.1.OR.NC1.GT.NCOL) THEN
          WRITE(IOUT,26) LINE
26          FORMAT(' Coordinates of the following record are ',
     &           'outside of the model grid:',/,A80)
          WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
          GO TO 10
        ENDIF
        LOC1=LOC+((KLAY-1)*NIJ+(NR1-1)*NCOL+NC1-1)*NW
        LOC2=LOC1
        LOC3=LOC1
        LOC4=LOC1
        W1=1.
        W2=0.
        W3=0.
        W4=0.
        IF(IBCHK) THEN
          IBLOC1=LCIBOU+(KLAY-1)*NIJ+(NR1-1)*NCOL+NC1-1
          IBLOC2=IBLOC1
          IBLOC3=IBLOC1
          IBLOC4=IBLOC1
        ELSE
          IBLOC1=0
          IBLOC2=0
          IBLOC3=0
          IBLOC4=0
        ENDIF
      ELSE IF(INTYP.EQ.'I') THEN
        NWT=4
        IF(NR2.LT.2.OR.NR2.GT.NROW.OR.NC2.LT.1.OR.NC2.GT.NCOL1) THEN
          WRITE(IOUT,26) LINE
          GO TO 10
        ENDIF
        LOC1=LOC+((KLAY-1)*NIJ+(NR2-1)*NCOL+NC2-1)*NW
        LOC2=LOC1+NW
        LOC3=LOC2-NCOL*NW
        LOC4=LOC3-NW
        CALL SHYDMW(XL,YL,X1,X2,Y1,Y2,W1,W2,W3,W4)
        IF(IBCHK) THEN
          IBLOC1=LCIBOU+(KLAY-1)*NIJ+(NR2-1)*NCOL+NC2-1
          IBLOC2=IBLOC1+1
          IBLOC3=IBLOC2-NCOL
          IBLOC4=IBLOC1-NCOL
        ELSE
          IBLOC1=0
          IBLOC2=0
          IBLOC3=0
          IBLOC4=0
        ENDIF
      ELSE
        WRITE(IOUT,27) LINE
27        FORMAT(' Invalid interpolation type was found on the ',
     &   'following record:',/,A80)
        WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
        GO TO 10
```

```
      ENDIF
      NUMH=NUMH+1
      HYDM(1,NUMH)=LOC1
      HYDM(2,NUMH)=LOC2
      HYDM(3,NUMH)=LOC3
      HYDM(4,NUMH)=LOC4
      HYDM(5,NUMH)=W1
      HYDM(6,NUMH)=W2
      HYDM(7,NUMH)=W3
      HYDM(8,NUMH)=W4
      HYDM(10,NUMH)=NW
      HYDM(12,NUMH)=IBLOC1
      HYDM(13,NUMH)=IBLOC2
      HYDM(14,NUMH)=IBLOC3
      HYDM(15,NUMH)=IBLOC4
      HYDM(16,NUMH)=KLAY
      HYDM(17,NUMH)=ITYP
      HYDM(18,NUMH)=NWT
      PTSLBL(NUMH)=HYDLBL
      IF(ARR.EQ.'DD') THEN
        IF(INTYP.EQ.'I')THEN
          H1=HOLD(NC2,NR2,KLAY)
          H2=HOLD(NC2+1,NR2,KLAY)
          H3=HOLD(NC2+1,NR2-1,KLAY)
          H4=HOLD(NC2,NR2-1,KLAY)
          HYDM(11,NUMH)=H1*W1+H2*W2+H3*W3+H4*W4
        ELSEIF(INTYP.EQ.'C')THEN
          HYDM(11,NUMH)=HOLD(NC1,NR1,KLAY)
        ENDIF
      ENDIF
      GO TO 10
 99   NNEW=NUMH-NUMHP
      IF(NNEW.GT.0)WRITE(IOUT,28) NNEW
 28   FORMAT(' A total of ',I3,' points have been added ',
     & 'for the hydrographs of BAS arrays.')
CC
      RETURN
      END
C=======================================================================
      SUBROUTINE HIBSRP(HYDM,NHYDM,NUMH,IHYDMUN,DELR,DELC,
     1NCOL,NROW,NLAY,LCIBOU,LCSUB,LCHC,IN,IOUT)
C
C
C-----VERSION 0100 29JULY1998 HIBSRP
C     ******************************************************************
C     READ SUBSIDENCE PACKAGE DATA FOR HYDROGRAPHS
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER HYDLBL*20,PTSLBL(5000)*20,LINE*80
      CHARACTER PCKG*3,ARR*2,INTYP*1
      DIMENSION HYDM(18,NHYDM),DELR(NCOL),DELC(NROW)
      LOGICAL IBCHK
      COMMON /HYDCOM/ PTSLBL
      COMMON /IBSCOM/ IBQ(200)
C     ------------------------------------------------------------------
      NIJ=NROW*NCOL
      NCOL1=NCOL-1
      NUMHP=NUMH
```

```
      REWIND(IN)
      READ(IN,'(A)',END=99) LINE
 10   READ(IN,'(A)',END=99) LINE
      LLOC=1
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      IF(LINE(ISTART:ISTOP).NE.'IBS') GO TO 10
      PCKG=LINE(ISTART:ISTOP)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      ARR=LINE(ISTART:ISTOP)
      WRITE(HYDLBL(1:2),FMT='(A2)')ARR
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      INTYP=LINE(ISTART:ISTOP)
      WRITE(HYDLBL(3:3),FMT='(A1)')INTYP
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,2,KLAY,R,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,XL,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,YL,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,0,N,R,IOUT,IN)
      WRITE(HYDLBL(4:6),FMT='(I3.3)')KLAY
      HYDLBL(7:20)=LINE(ISTART:ISTART+14)
cc    TIME SERIES FROM THE CRITICAL HEAD ARRAY
      IF (ARR.EQ.'HC') THEN
        LOC=LCHC
        NW=1
        ITYP=1
        IBCHK=.TRUE.
cc    TIME SERIES FROM THE COMPACTION ARRAY
      ELSE IF (ARR.EQ.'CP') THEN
        LOC=LCSUB
        NW=1
        ITYP=1
        IBCHK=.TRUE.
cc    TIME SERIES FROM THE SUBSIDENCE ARRAY
      ELSE IF (ARR.EQ.'SB') THEN
        LOC=LCSUB
        NW=1
        ITYP=3
        NQ=0
        IBCHK=.FALSE.
C ----- SET THE CORRECT LAYER COUNTER FOR SUBSIDENCE ARRAYS
        DO 20 K=1,KLAY
        IF(IBQ(K).GT.0) NQ=NQ+1
 20     CONTINUE
        KLAY=NQ
      ELSE
        WRITE(IOUT,25) LINE
 25     FORMAT(' Invalid array type was found on the following',
     &   ' record:',/,A80)
        WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
        GO TO 10
      ENDIF
      CALL GRDLOC(XL,YL,DELR,DELC,NROW,NCOL,NR1,NC1,NR2,NC2,
     & X1,X2,Y1,Y2)
      IF(INTYP.EQ.'C') THEN
        NWT=1
        IF(NR1.LT.1.OR.NR1.GT.NROW.OR.NC1.LT.1.OR.NC1.GT.NCOL) THEN
          WRITE(IOUT,26) LINE
 26       FORMAT(' Coordinates of the following record are ',
     &         'outside of the model grid:',/,A80)
          WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
          GO TO 10
```

```
      ENDIF
      LOC1=LOC+((KLAY-1)*NIJ+(NR1-1)*NCOL+NC1-1)*NW
      LOC2=LOC1
      LOC3=LOC1
      LOC4=LOC1
      W1=1.
      W2=0.
      W3=0.
      W4=0.
      IF(IBCHK) THEN
        IBLOC1=LCIBOU+(KLAY-1)*NIJ+(NR1-1)*NCOL+NC1-1
        IBLOC2=IBLOC1
        IBLOC3=IBLOC1
        IBLOC4=IBLOC1
      ELSE
        IBLOC1=0
        IBLOC2=0
        IBLOC3=0
        IBLOC4=0
      ENDIF
    ELSE IF(INTYP.EQ.'I') THEN
      NWT=4
      IF(NR2.LT.2.OR.NR2.GT.NROW.OR.NC2.LT.1.OR.NC2.GT.NCOL1) THEN
        WRITE(IOUT,26) LINE
        GO TO 10
      ENDIF
      LOC1=LOC+((KLAY-1)*NIJ+(NR2-1)*NCOL+NC2-1)*NW
      LOC2=LOC1+NW
      LOC3=LOC2-NCOL*NW
      LOC4=LOC3-NW
      CALL SHYDMW(XL,YL,X1,X2,Y1,Y2,W1,W2,W3,W4)
      IF(IBCHK) THEN
        IBLOC1=LCIBOU+(KLAY-1)*NIJ+(NR1-1)*NCOL+NC1-1
        IBLOC2=IBLOC1+1
        IBLOC3=IBLOC2-NCOL
        IBLOC4=IBLOC1-NCOL
      ELSE
        IBLOC1=0
        IBLOC2=0
        IBLOC3=0
        IBLOC4=0
      ENDIF
    ELSE
      WRITE(IOUT,27) LINE
27     FORMAT(' Invalid interpolation type was found on the ',
     &  'following record:',/,A80)
      WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
      GO TO 10
    ENDIF
    NUMH=NUMH+1
    HYDM(1,NUMH)=LOC1
    HYDM(2,NUMH)=LOC2
    HYDM(3,NUMH)=LOC3
    HYDM(4,NUMH)=LOC4
    HYDM(5,NUMH)=W1
    HYDM(6,NUMH)=W2
    HYDM(7,NUMH)=W3
    HYDM(8,NUMH)=W4
    HYDM(10,NUMH)=NW
    HYDM(12,NUMH)=IBLOC1
```

```
      HYDM(13,NUMH)=IBLOC2
      HYDM(14,NUMH)=IBLOC3
      HYDM(15,NUMH)=IBLOC4
      HYDM(16,NUMH)=KLAY
      HYDM(17,NUMH)=ITYP
      HYDM(18,NUMH)=NWT
      PTSLBL(NUMH)=HYDLBL
      GO TO 10
 99   NNEW=NUMH-NUMHP
      IF(NNEW.GT.0)WRITE(IOUT,28) NNEW
 28   FORMAT(' A total of ',I3,' points have been added ',
     & 'for the hydrographs of IBS arrays.')
      RETURN
      END
CC

C=======================================================================
      SUBROUTINE HSTRRP(X,ISUM,HYDM,NHYDM,NUMH,IHYDMUN,DELR,DELC,
     1NCOL,NROW,NLAY,LCIBOU,LCSTRM,ICSTRM,NSTREM,IN,IOUT)
C
C
C-----VERSION 0100 29JULY1998 HSTRRP
C     ******************************************************************
C     READ STREAM PACKAGE DATA FOR HYDROGRAPHS
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER HYDLBL*20,PTSLBL(5000)*20,LINE*80
      CHARACTER PCKG*3,ARR*2,INTYP*1
      DIMENSION X(ISUM),HYDM(18,NHYDM),DELR(NCOL),DELC(NROW)
      LOGICAL IBCHK
      COMMON /HYDCOM/ PTSLBL
C     ------------------------------------------------------------------
      NIJ=NROW*NCOL
      NCOL1=NCOL-1
      NUMHP=NUMH
      REWIND(IN)
      READ(IN,'(A)',END=99) LINE
 10   READ(IN,'(A)',END=99) LINE
      LLOC=1
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      IF(LINE(ISTART:ISTOP).NE.'STR') GO TO 10
      PCKG=LINE(ISTART:ISTOP)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      ARR=LINE(ISTART:ISTOP)
      WRITE(HYDLBL(1:2),FMT='(A2)')ARR
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,1,N,R,IOUT,IN)
      INTYP=LINE(ISTART:ISTOP)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,2,KLAY,R,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,XL,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,3,N,YL,IOUT,IN)
      CALL URWORD(LINE,LLOC,ISTART,ISTOP,0,N,R,IOUT,IN)
      WRITE(HYDLBL(3:5),FMT='(I3.3)')INT(XL)
      WRITE(HYDLBL(6:8),FMT='(I3.3)')INT(YL)
      HYDLBL(9:20)=LINE(ISTART:ISTART+12)
cc    XL contains the SEGMENT number and YL contains the REACH number.
cc    ARR/IRR designate the streamflow option with
cc    ST==>IRR=6 (stream stage), SO==>IRR=7 (out of reach),
cc    SI==>IRR=8 (into reach), and SA==>IRR=9(into aquifer)
```

```
cc   TIME SERIES FROM THE STREAM STAGE OF RIVER IN CELL
     IF (ARR.EQ.'ST') THEN
       LOC=LCSTRM
       NW=1
       ITYP=1
       IRR=6
       IBCHK=.FALSE.
cc   TIME SERIES FROM THE STREAMFLOW OUT OF CELL
     ELSE IF (ARR.EQ.'SO') THEN
       LOC=LCSTRM
       NW=1
       ITYP=1
       IRR=7
       IBCHK=.FALSE.
cc   TIME SERIES FROM THE STREAMFLOW INTO CELL
     ELSE IF (ARR.EQ.'SI') THEN
       LOC=LCSTRM
       NW=1
       ITYP=1
       IRR=8
       IBCHK=.FALSE.
cc   TIME SERIES FROM THE INTO OR OUT OF AQUIFER
     ELSE IF (ARR.EQ.'SA') THEN
       LOC=LCSTRM
       NW=1
       ITYP=1
       IRR=9
       IBCHK=.TRUE.
     ELSE IF(NSTREM.LT.1)THEN
     WRITE(IOUT,24) LINE
 24    FORMAT(' No Active Streams in this Model:',/,A80)
       WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
       GO TO 10
     ELSE
       WRITE(IOUT,25) LINE
 25     FORMAT(' Invalid streamflow feature was found on the following',
    &   ' record:',/,A80)
       WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
       GO TO 10
     ENDIF
C
      K=0
 26    KK=5*K
      KS=KK+3
      KR=KK+4
      IKS=ICSTRM+KS
      IKR=ICSTRM+KR
      ISEG=ICONV(X(IKS))
      IRCH=ICONV(X(IKR))
      IF(ISEG.EQ.INT(XL).AND.IRCH.EQ.INT(YL))THEN
       INR=ICSTRM+KK+1
       INC=ICSTRM+KK+2
       NR1=ICONV(X(INR))
       NC1=ICONV(X(INC))
       ISTRM=K
      ELSEIF(K.LT.NSTREM)THEN
       K=K+1
       GOTO 26
      ENDIF
     IF(INTYP.EQ.'C') THEN
```

```
      NWT=1
      IF(NR1.LT.1.OR.NR1.GT.NROW.OR.NC1.LT.1.OR.NC1.GT.NCOL) THEN
        WRITE(IOUT,27) LINE
 27       FORMAT(' Coordinates of the following record are ',
     &          'outside of the model grid:',/,A80)
        WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
        GO TO 10
      ENDIF
      LOC1=LOC+(ISTRM*11)+IRR+1
      LOC2=LOC1
      LOC3=LOC1
      LOC4=LOC1
      W1=1.
      W2=0.
      W3=0.
      W4=0.
      IF(IBCHK) THEN
        IBLOC1=LCIBOU+(KLAY-1)*NIJ+(NR1-1)*NCOL+NC1-1
        IBLOC2=IBLOC1
        IBLOC3=IBLOC1
        IBLOC4=IBLOC1
      ELSE
        IBLOC1=0
        IBLOC2=0
        IBLOC3=0
        IBLOC4=0
      ENDIF
    ELSE
      WRITE(IOUT,28) LINE
 28     FORMAT(' Invalid interpolation type was found on the ',
     &   'following record:',/,A80)
      WRITE(IOUT,*) 'Hydrograph Record will be ignored.'
      GO TO 10
    ENDIF
    NUMH=NUMH+1
    HYDM(1,NUMH)=LOC1
    HYDM(2,NUMH)=LOC2
    HYDM(3,NUMH)=LOC3
    HYDM(4,NUMH)=LOC4
    HYDM(5,NUMH)=W1
    HYDM(6,NUMH)=W2
    HYDM(7,NUMH)=W3
    HYDM(8,NUMH)=W4
    HYDM(10,NUMH)=NW
    HYDM(12,NUMH)=IBLOC1
    HYDM(13,NUMH)=IBLOC2
    HYDM(14,NUMH)=IBLOC3
    HYDM(15,NUMH)=IBLOC4
    HYDM(16,NUMH)=KLAY
    HYDM(17,NUMH)=ITYP
    HYDM(18,NUMH)=NWT
    PTSLBL(NUMH)=HYDLBL
    GO TO 10
 99   NNEW=NUMH-NUMHP
    IF(NNEW.GT.0)WRITE(IOUT,29) NNEW
 29   FORMAT(' A total of ',I3,' points have been added ',
    & 'for the hydrographs of STR arrays.')
CC
    RETURN
    END
```

Output Subroutine

```
      SUBROUTINE HYDMOT(X,ISUM,HYDM,NUMH,IHYDMUN,TOTIM,HYDNOH,
     1 NROW,NCOL,ITMUNI,IOUT)
C
C
C-----VERSION 0100 29JULY1998 HYDMOT
C     ******************************************************************
C     WRITE HYDROGRAPH RECORDS
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER TIMLBL*4,PTSLBL(5000)*20
      LOGICAL*4 FIRST
      SAVE FIRST
      DIMENSION X(ISUM),HYDM(18,NUMH)
      COMMON /HYDCOM/ PTSLBL
      DATA FIRST/.TRUE./
C     ------------------------------------------------------------------
C
C -----If this is the first time into the output routine, write header
C -----records
C ------WRITE HYDROGRAPH HEADER RECORD
      IF(FIRST) THEN
      TIMLBL='TIME'
      FIRST=.FALSE.
C
        IF(NUMH.LE.0) THEN
          WRITE(IOUT,120)
 120        FORMAT(1X,'HYDROGRAPH OPTION CANCELLED BECAUSE NO VALID ',
     1     'HYDROGRAPH POINTS EXIST.')
          RETURN
        ENDIF
C
        WRITE(IHYDMUN) NUMH,ITMUNI
        WRITE(IOUT,130) NUMH
 130     FORMAT(1X,'A TOTAL OF ',I3,' HYDROGRAPH POINTS HAVE BEEN ',
     1      'PREPARED.')
C ------WRITE HYDROGRAPH LABEL HEADER RECORD
        WRITE(IHYDMUN) TIMLBL,(PTSLBL(N),N=1,NUMH)
      ENDIF
C
C -----Calculate the number of cells in one layer of the grid.
      NIJ=NROW*NCOL
      IF(NUMH.LE.0)RETURN
C
C -----Calculate value for each hydrograph point.
      DO 50 N=1,NUMH
      NN=N
C -----Determine interpolation type, word length, number of weights,
C -----and locations of IBOUND codes at or around hydrograph point.
      ITYP=HYDM(17,N)
      NW = HYDM(10,N)
      NWT= HYDM(18,N)
      IBLOC1=HYDM(12,N)
      IBLOC2=HYDM(13,N)
      IBLOC3=HYDM(14,N)
      IBLOC4=HYDM(15,N)
```

```
C -----If IBOUND is to be checked for inactive cells, retrieve values
C -----at or around hydrograph point.
      IF(IBLOC1.GT.0) THEN
        IB1=ICONV(X(IBLOC1))
        IBFACT=IB1
        IF(NWT.EQ.4) THEN
          IB2=ICONV(X(IBLOC2))
          IB3=ICONV(X(IBLOC3))
          IB4=ICONV(X(IBLOC4))
          IBFACT=IB1*IB2*IB3*IB4
        ENDIF
      ENDIF
C -----Compute hydrograph value by interpolation of one or more
C -----values in the X array.
      IF(ITYP.EQ.1) THEN
        IF(IBLOC1.NE.0.AND.IBFACT.EQ.0) THEN
          HYDM(9,N)=HYDNOH
        ELSE
          HYDM(9,N)=WTAVG(X,HYDM,ISUM,NUMH,NN,NW,NWT,0)
        ENDIF
C -----Compute hydrograph value as the difference between a constant and
C -----a value interpolated from one or more values in the X array.
      ELSE IF(ITYP.EQ.2) THEN
        IF(IBLOC1.NE.0.AND.IBFACT.EQ.0) THEN
          HYDM(9,N)=HYDNOH
        ELSE
          HYDM(9,N)=HYDM(11,N)-WTAVG(X,HYDM,ISUM,NUMH,NN,NW,NWT,0)
        ENDIF
C -----Compute hydrograph value as the sum of interpolated values in
C -----arrays from one or more contiguous model layers.
      ELSE IF(ITYP.EQ.3) THEN
        KLAY=HYDM(16,N)
        TOTL=0.0
        DO 9 K=1,KLAY
        NOFF=-NIJ*(KLAY-K)
        TOTL=TOTL+WTAVG(X,HYDM,ISUM,NUMH,NN,NW,NWT,NOFF)
    9   CONTINUE
        HYDM(9,N)=TOTL
      ENDIF
   50 CONTINUE
C
C ------WRITE HYDROGRAPH RECORD
      WRITE(IHYDMUN) TOTIM,(HYDM(9,N),N=1,NUMH)
C
C ------ RETURN
      RETURN
      END
```

```
      SUBROUTINE GRDLOC(XL,YL,DELR,DELC,NROW,NCOL,NR1,NC1,NR2,NC2,
     & XX1,XX2,YY1,YY2)
C
C
C-----VERSION 0100 29JULY1998 GRDLOC
C     ****************************************************************
C     LOCATE CELLS FOR HYDROGRAPH POINTS
C     ****************************************************************
C
C     SPECIFICATIONS:
C     ----------------------------------------------------------------
      DIMENSION DELR(NCOL),DELC(NROW)
C     ----------------------------------------------------------------
      XX1=0.
      XX2=0.
      YY1=0.
      YY2=0.
      X1=0.
      IF(XL.LT.X1) THEN
        NC1=0
        NC2=0
        GO TO 100
      ENDIF
      XCB=0.
      XCF=DELR(1)*0.5
      DO 10 N=1,NCOL
      X2=X1+DELR(N)
      XC=XCF
      DXF=0.
      IF(N.LT.NCOL) DXF=DELR(N+1)*0.5
      XCF=X2+DXF
      IF(XL.LE.X2) THEN
        NC1=N
        IF(XL.LT.XC) THEN
          NC2=N-1
          XX1=XCB
          XX2=XC
        ELSE
          NC2=N
          XX1=XC
          XX2=XCF
        ENDIF
        GO TO 100
      ENDIF
      X1=X2
      XCB=XC
 10   CONTINUE
      NC1=NCOL+1
      NC2=NCOL+1
 100  Y1=0.
      YCB=0.
      YCF=DELC(NROW)*0.5
      IF(YL.LT.Y1) THEN
        NR1=NROW+1
        NR2=NROW+1
        RETURN
      ENDIF
```

```
      DO 110 N=NROW,1,-1
      Y2=Y1+DELC(N)
      YC=YCF
      DYF=0.
      IF(N.GT.1) DYF=DELC(N-1)*0.5
      YCF=Y2+DYF
      IF(YL.LE.Y2) THEN
        NR1=N
        IF(YL.LT.YC) THEN
          NR2=N+1
          YY1=YCB
          YY2=YC
        ELSE
          NR2=N
          YY1=YC
          YY2=YCF
        ENDIF
        RETURN
      ENDIF
      Y1=Y2
      YCB=YC
  110 CONTINUE
      NC1=0
      NC2=0
      RETURN
      END

C=======================================================================
      FUNCTION WTAVG(X,HYDM,ISUM,NUMH,N,NW,NWT,NOFF)
C
C
C-----VERSION 0100 26MAY1998 WTAVG
C     ***************************************************************
C     COMPUTE WEIGHTED AVERAGE
C     ***************************************************************
C
C     SPECIFICATIONS:
C     ---------------------------------------------------------------
      DIMENSION X(ISUM),HYDM(18,NUMH)
C     ---------------------------------------------------------------

      WTAVG=0.0
      DO 10 M=1,NWT
      LOC=HYDM(M,N)+NOFF
      IF(NW.EQ.2) THEN
        ZZ=RCONV(X(LOC))
      ELSE
        ZZ=X(LOC)
      ENDIF
      WTAVG=WTAVG+ZZ*HYDM(4+M,N)
   10 CONTINUE
      RETURN
      END
```

```
C=======================================================================
      FUNCTION RCONV(X1)
C
C-----VERSION 0100 15APR1991 RCONV
C     *****************************************************************
C     TRANSLATE DOUBLE PRECISION VALUES FROM X VECTOR TO SINGLE
C     PRECISION VALUES
C     *****************************************************************
C
C       SPECIFICATIONS:
C     -----------------------------------------------------------------
      DOUBLE PRECISION X1
C
      RCONV=SNGL(X1)
C
C ------RETURN
      RETURN
      END
C=======================================================================
      INTEGER FUNCTION ICONV(II)
C
C-----VERSION 0100 16MAR1998 ICONV
C     *****************************************************************
C     TRANSLATE INTEGER VALUES STORED IN X VECTOR TO INTEGER
C     VALUES IN HYDMOD SUBROUTINES
C     *****************************************************************
C
C       SPECIFICATIONS:
C     -----------------------------------------------------------------
C
      ICONV=II
      RETURN
      END
C=======================================================================
      SUBROUTINE SHYDMW(X0,Y0,X1,X2,Y1,Y2,W1,W2,W3,W4)
C
C-----VERSION 0100 15APR1991 SHYDMW
C     *****************************************************************
C     COMPUTE WEIGHTS FOR BILINEAR INTERPOLATION
C     *****************************************************************
C
C       SPECIFICATIONS:
C     -----------------------------------------------------------------
C
      DX=(X0-X1)/(X2-X1)
      DY=(Y0-Y1)/(Y2-Y1)
      DXY=DX*DY
      W1=1-DX-DY+DXY
      W2=DX-DXY
      W3=DXY
      W4=DY-DXY
      RETURN
      END
```

REFERENCES CITED

Harbaugh, A.W., and McDonald, M.G., 1996a, User's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 96-485, 56 p.

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

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

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

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

Appendix 1: input data sets and selected output for the Test Problem.

A simulation was run to exemplify the application of selected features of the *HYDMOD* program. Details of the test problem and results are discussed in the section "Test Problem" and are shown in figures 2 and 3.

INPUT DATA SETS FOR TEST PROBLEM

Basic Package Input Data Set

The input for the Basic (BAS) Package follows the column numbers. The input consists of 43 records, all of which are shown below. Input for the BAS package is read from the unit number for Basic Package input specified in the **MAIN** program.

```
                                    Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Modflow-96 Hydrograph Package (HYDMOD)Example Data Set
Hanson,RT,and Leake,SA,1999, Open-File Report 98-564
         3         5         6         1         4
 11 15  0  0  0  0  0 12 13  0  0 14  0  0  0  0  0 16 17  0  0 18
         0         1         IAPART,ISTRT
         5         1(80I2)
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 0
 1 1 1 1 0 0
-1-1-1 0 0 0
         5         1(80I2)
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 0
 1 1 1 1 0 0
 1 1 1 0 0 0
         5         1(80I2)
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 0
 1 1 1 1 0 0
```

Input for the Basic Package is continued following the column numbers.

```
                                    Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      999.        HNOFLO
         5       1.0(10F8.2)
  383.47  384.17  386.37  391.45  397.39  400.16
  377.74  378.57  381.25  388.31  395.64  399.53
  369.42  370.32  373.66  385.10  395.15  999.00
  360.13  360.78  363.84  382.21  999.00  999.00
  350.00  350.00  350.00  999.00  999.00  999.00
         5       1.0(10F8.2)
  383.90  384.08  384.63  385.78  387.46  388.92
  383.37  383.56  384.14  385.41  387.25  388.88
  382.76  382.95  383.54  384.98  387.00  999.00
  382.30  382.46  382.98  384.59  999.00  999.00
  382.01  382.14  382.47  999.00  999.00  999.00
         5       1.0(10F8.2)
  385.48  385.49  385.51  385.57  385.66  385.75
  385.47  385.48  385.50  385.56  385.65  385.75
  385.46  385.46  385.49  385.54  385.63  999.00
  385.45  385.46  385.48  385.53  999.00  999.00
  385.45  385.45  385.46  999.00  999.00  999.00
    1000.        20       1.2
```

Block-Centered Flow Package Input Data Set

The input for the Block-Centered Flow Package follows the column numbers. The input consists of 15 records, all of which are shown below. Input for this package is read from the unit number 11 as specified in the input data set for the Basic Package.

```
                                    Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         0         0                                          ISS, IBCFCB
 0 0 0
         0        1.                                        TRPY
        11     1000.(16F5.0)                               DELR
   .5   .7   .9  1.2  1.5  1.8
        11     1000.(16F5.0)                               DELC
  1.   .8   .6   .5   .5
         0      0.10                                         SF1 - layer 1
         0     2000.                                       tran1 - layer 1
         0     1.E-4                                      vcont1 - layer 1
         0     5.E-5                                        SF2 - layer 2
         0     1000.                                       tran2 - layer 2
         0     1.E-5                                      vcont2 - layer 2
         0     1.E-5                                        SF3 - layer 3
         0     1000.                                       tran3 - layer 3
```

Strongly Implicit Procedure Package Input Data Set

The input for the Strongly Implicit Procedure Package follows the column numbers. The input consists of two records, all of which are shown below. Input for this package is read from unit number 13 as specified in the input data set for the Basic Package.

```
                                    Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
       120         5   MXITER, NPARM
        1.    .00100         1         0         5
```

Recharge Package Input Data Set

The input for the Recharge Package follows the column numbers. The input consists of three records, all of which are shown below. Input for this package is read from unit number 12 as specified in the input data set for the Basic Package.

```
                                    Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        1         0
        1         0
        0      .008
```

Well Package Input Data Set

The input for the Well Package follows the column numbers. The input consists of four records, all of which are shown below. Input for this package is read from unit number 15 as specified in the input data set for the Basic Package.

```
                                    Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        2         0
        2
        1         1         1   -80000.
        2         1         1   -55000.
```

Output Control Package Input Data Set

The input for the Output Control Package follows the column numbers. The input consists of 23 records, all of which are shown below. Input for this package is read from unit number 14 as specified in the input data set for the Basic Package.

```
                                  Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        4         4        21        22
        0         1         0         1
        0         0         1         0
       -1         1         0         0                   time-step  2
       -1         1         0         0                   time-step  3
       -1         1         0         0                   time-step  4
       -1         1         0         0                   time-step  5
       -1         1         0         0                   time-step  6
       -1         1         0         0                   time-step  7
       -1         1         0         0                   time-step  8
       -1         1         0         0                   time-step  9
       -1         1         0         0                   time-step 10
       -1         1         0         0                   time-step 11
       -1         1         0         0                   time-step 12
       -1         1         0         0                   time-step 13
       -1         1         0         0                   time-step 14
       -1         1         0         0                   time-step 15
       -1         1         0         0                   time-step 16
       -1         1         0         0                   time-step 17
       -1         1         0         0                   time-step 18
       -1         1         0         0                   time-step 19
        0         1         1         1                   time-step 20
```

Streamflow-Routing Package Input Data Set

The input for the Streamflow-Routing Package follows the column numbers. The input consists of 25 records, all of which are shown below. Input for this package is read from unit number 16 as specified in the input data set for the Basic Package.

```
                                        Column Numbers
       1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      10         3         2         0         1  128390.4        -1         0
      10         0         0
   1    1    6    1    1       8640000.0     395.0    1000.00     390.0     392.0
   1    2    6    1    2                     390.0     800.00     385.0     387.0
   1    2    5    1    3                     387.0     800.00     383.0     385.0
   1    2    4    1    4                     386.0     800.00     382.0     384.0
   1    2    3    1    5                     384.0     600.00     380.0     382.0
   1    1    3    2    1        864000.0     386.0     400.00     381.0     383.0
   1    2    3    2    2                     382.0     400.80     377.0     379.0
   1    2    3    3    1            -1.0     378.0     400.00     373.0     375.0
   1    3    2    3    2                     375.0     400.00     370.0     372.0
   1    4    1    3    3                     373.0     400.00     370.0     371.0
      50.     0.007     0.030
      50.     0.007     0.030
      50.     0.002     0.022
      50.     0.002     0.022
      50.     0.002     0.022
       5.     0.004     0.022
      10.     0.005     0.030
      60.     0.005     0.030
      70.     0.005     0.030
      80.     0.005     0.030
   0    0
   0    0
   1    2
```

Interbed-Storage Package Input Data Set

The input for the Interbed-Storage Package follows the column numbers. The input consists of 29 records, all of which are shown below. Input for this package is read from unit number 17 as specified in the input data set for the Basic Package.

```
                                    Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        0         0
 1 1 1
       17       1.(6F8.2)                   -1      LAYER 1 CRITICAL HEADS
  383.47  384.17  386.37  391.45  397.39  400.16
  377.74  378.57  381.25  388.31  395.64  399.53
  369.42  370.32  373.66  385.10  395.15  999.00
  360.13  360.78  363.84  382.21  999.00  999.00
  350.00  350.00  350.00  999.00  999.00  999.00
       0    .0001                          LAYER 1 ELASTIC STORAGE COEFFICIENT
       0     .001                          LAYER 1 INELASTIC STORAGE COEFFICIENT
       0      0.0                          LAYER 1 INELASTIC STORAGE COEFFICIENT
      17       1.(6F8.2)                   -1      LAYER 2 CRITICAL HEADS
  383.90  384.08  384.63  385.78  387.46  388.92
  383.37  383.56  384.14  385.41  387.25  388.88
  382.76  382.95  383.54  384.98  387.00  999.00
  382.30  382.46  382.98  384.59  999.00  999.00
  382.01  382.14  382.47  999.00  999.00  999.00
       0    .0001                          LAYER 2 ELASTIC STORAGE COEFFICIENT
       0     .001                          LAYER 2 INELASTIC STORAGE COEFFICIENT
       0      0.0                          LAYER 2 PREVIOUS COMPACTION
      17       1.(6F8.2)                   -1      LAYER 3 CRITICAL HEADS
  385.48  385.49  385.51  385.57  385.66  385.75
  385.47  385.48  385.50  385.56  385.65  385.75
  385.46  385.46  385.49  385.54  385.63  999.00
  385.45  385.46  385.48  385.53  999.00  999.00
  385.45  385.45  385.46  999.00  999.00  999.00
       0    .0001                          LAYER 3 ELASTIC STORAGE COEFFICIENT
       0     .001                          LAYER 3 INELASTIC STORAGE COEFFICIENT
       0      0.0                          LAYER 3 PREVIOUS COMPACTION
```

Hydrograph Program Input Data Set

The input for the Hydrograph Program follows the column numbers. The input consists of 17 records, all of which are shown below. Input for this program is read from unit number 18 as specified in the input data set for the Basic Package.

```
                                        Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        16        33     998.0
BAS     DD  C  1     250.0     2900. 'PUMP_WELL'
BAS     DD  C  2     250.0     2900. 'PUMP_WELL'
BAS     DD  C  3     250.0     2900. 'PUMP_WELL'
BAS     HD  I  1    1400.0     2200. 'USGS_OB_No_11'
BAS     HD  I  2    1400.0     2200. 'USGS_OB_No_12'
BAS     HD  I  3    1400.0     2200. 'USGS_OB_No_13'
BAS     HD  I  1    2200.0     1200. 'USGS_OB_No_21'
BAS     HD  I  2    2200.0     1200. 'USGS_OB_No_22'
BAS     HD  I  3    2200.0     1200. 'USGS_OB_No_23'
BAS     HD  C  1    3000.0      700. 'USGS_OB_No_31'
BAS     HD  C  2    3000.0      700. 'USGS_OB_No_32'
BAS     HD  C  3    3000.0      700. 'USGS_OB_No_33'
IBS     CP  I  2     600.0     2500. 'USGS_EXTENSOMETER'
IBS     SB  I  3     600.0     2500. 'USGS_BENCHMARK'
STR     SO  C  1        3.        2. 'USGS_STREAM_GAGE'
STR     SA  C  1        3.        2. 'STREAM_LOSS_GAGE'
```

SELECTED OUTPUT DATA SETS FOR TEST PROBLEM

Hydrograph Program Output Data Sets

Selected output for the Hydrograph program follows the column numbers. The output consists of separate files that each contain 22 records, all of which are shown below and on the following pages. Output from this program is read by the Hydrograph program post-processor program HYDFMT, which generates separate files for each data type specified by the user in the input file for the Hydrograph program and selectively requested with the post-processor program.

Columnar output for drawdown at pumped well

Column Numbers

```
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
  3 TIME              DDC001PUMP_WELL      DDC002PUMP_WELL      DDC003PUMP_WELL
 1998.000                  0.00                 0.00                 0.00
 1998.015                  6.26                51.25                 0.91
 1998.032                 11.75                65.64                 2.17
 1998.053                 16.61                72.80                 3.85
 1998.079                 21.00                77.15                 5.91
 1998.109                 25.04                80.06                 8.35
 1998.146                 28.81                82.23                11.13
 1998.190                 32.36                84.08                14.21
 1998.242                 35.73                85.81                17.53
 1998.305                 38.92                87.55                21.05
 1998.381                 41.95                89.31                24.69
 1998.472                 44.79                91.11                28.37
 1998.581                 47.46                92.94                31.99
 1998.712                 49.93                94.75                35.48
 1998.869                 52.21                96.51                38.75
 1999.057                 54.27                98.18                41.73
 1999.283                 56.09                99.71                44.37
 1999.555                 57.67               101.07                46.62
 1999.880                 59.00               102.26                48.51
 2000.271                 60.09               103.28                50.06
 2000.740                 60.98               104.15                51.31
```

Part of columnar output for heads at observation-well sites

```
                                  Column Numbers
       1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
  9 TIME             HDI001USGS_OB_No_11  HDI002USGS_OB_No_12  HDI003USGS_OB_No_13
1998.000               381.58               384.07               385.50
1998.015               381.36               366.85               384.79
1998.032               380.88               356.16               383.59
1998.053               380.12               349.73               381.95
1998.079               379.09               345.69               379.90
1998.109               377.81               342.99               377.47
1998.146               376.30               341.00               374.69
1998.190               374.60               339.32               371.62
1998.242               372.74               337.73               368.29
1998.305               370.78               336.12               364.77
1998.381               368.76               334.47               361.14
1998.472               366.69               332.76               357.46
1998.581               364.65               331.02               353.84
1998.712               362.65               329.26               350.35
1998.869               360.75               327.55               347.08
1999.057               358.99               325.92               344.10
1999.283               357.39               324.41               341.47
1999.555               355.99               323.07               339.21
1999.880               354.79               321.89               337.33
2000.271               353.78               320.87               335.78
2000.740               352.95               320.00               334.53
```

Columnar output for streamflow at gaging station

```
                                  Column Numbers
       1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1 TIME             SO003002USGS_STREAM_
1998.000                    0.00
1998.015              9533554.00
1998.032              9532694.00
1998.053              9531393.00
1998.079              9529512.00
1998.109              9527407.00
1998.146              9524908.00
1998.190              9522548.00
1998.242              9520145.00
1998.305              9517276.00
1998.381              9514387.00
1998.472              9511851.00
1998.581              9509503.00
1998.712              9506818.00
1998.869              9503868.00
1999.057              9500775.00
1999.283              9497697.00
1999.555              9494979.00
1999.880              9493254.00
2000.271              9492128.00
2000.740              9491399.00
```

Columnar output for streamflow infiltration at gaging station

Column Numbers

```
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1 TIME              SA003002STREAM_LOSS_
1998.000                 0.00
1998.015               954.13
1998.032              1047.74
1998.053              1047.73
1998.079              1047.69
1998.109              1047.67
1998.146              1047.62
1998.190              1047.58
1998.242              1047.55
1998.305              1047.51
1998.381              1047.46
1998.472              1047.42
1998.581              1047.39
1998.712              1047.34
1998.869              1047.30
1999.057              1047.25
1999.283              1047.20
1999.555              1047.16
1999.880              1047.13
2000.271              1047.12
2000.740              1047.11
```

Columnar output for compaction at extensometer well

Column Numbers

```
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1 TIME              CPI002USGS_EXTENSOME
1998.000                 0.00
1998.015                 0.03
1998.032                 0.04
1998.053                 0.05
1998.079                 0.06
1998.109                 0.06
1998.146                 0.06
1998.190                 0.06
1998.242                 0.06
1998.305                 0.07
1998.381                 0.07
1998.472                 0.07
1998.581                 0.07
1998.712                 0.07
1998.869                 0.07
1999.057                 0.08
1999.283                 0.08
1999.555                 0.08
1999.880                 0.08
2000.271                 0.08
2000.740                 0.08
```

Columnar output for subsidence of bench mark at extensometer well

Column Numbers

```
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
  1 TIME              SBI003USGS_BENCHMARK
1998.000                  0.00
1998.015                  0.03
1998.032                  0.05
1998.053                  0.06
1998.079                  0.07
1998.109                  0.08
1998.146                  0.09
1998.190                  0.09
1998.242                  0.10
1998.305                  0.11
1998.381                  0.12
1998.472                  0.13
1998.581                  0.13
1998.712                  0.14
1998.869                  0.15
1999.057                  0.15
1999.283                  0.16
1999.555                  0.17
1999.880                  0.17
2000.271                  0.17
2000.740                  0.18
```

Appendix 2: *HYDMOD* post-processor program HYDFMT.

HYDFMT is a simple post-processor program that allows extraction of time-series of model-computed head, drawdown, and other variables for selected model cells or well sites that are saved by the *HYDMOD* program (Steps 5, 6, fig. 1). Program HYDFMT, presented here, translates the *HYDMOD* binary output file into multiple ASCII-text files that can be used for additional analyses or graphical presentation of simulation results. The program is a simple way of extracting hydrograph information from unformatted output data that were recorded during a *MODFLOW* simulation and writes all or selected columns of data into multiple formatted files. These formatted files can then be used with graphics programs, spreadsheets, data bases, or other analysis programs to evaluate the performance of the *MODFLOW* simulation.

USE OF THE HYDFMT PROGRAM

Prior to using HYDFMT, a user must have run *MODFLOW* and have saved time-series information at user-specified locations from *MODFLOW* in an unformatted file at all time steps using the *HYDMOD* program. Output from HYDFMT includes a file containing time values and a series of corresponding data values for specified model locations of interest. Additionally, a header record is written to the output file that contains the number of time-series retrieved from the unformatted hydrograph output file, a label for the time column, and labels for columns of data for the individual specified locations.

Input to HYDFMT is interactive. Users must specify the name of the unformatted file from which hydrograph values will be extracted, a root name used as a prefix for the ASCII-text files created by HYDFMT, and parameters for transforming model simulation time into other time units. The HYDFMT queries and example user responses are shown in figure 4.

The unformatted file created by the *HYDMOD* program can include data for locations in any model layer and can be any combination of data types. Output from HYDFMT, in contrast, is segregated into separate output files for each data type specified by the ARR variable in the *HYDMOD* input file. Depending on which data types are specified when *HYDMOD* is run with *MODFLOW,* HYDFMT will create up to nine separate columnar ASCII-text files for water-level altitude, drawdown, critical head, compaction, subsidence, stage of streamflow in a reach (cell), streamflow into a reach (cell), streamflow out of a reach (cell), and flow between streams and the ground water in the adjacent aquifer.

The time stored with each unformatted array recorded by *MODFLOW* is the elapsed time since the start of the simulation. The units of time recorded are as specified by the ITMUNI variable in the *MODFLOW* Basic Package input file (McDonald and Harbaugh, 1988). To convert simulation time to real-world time in the HYDFMT output, users can specify either a calendar-type date format of year, month, and day of month (YYYYMMDD) or a decimal time format. In the example shown in figure 4, model simulation time is in days, but time values in decimal years, including fractions, are desired for hydrograph output. The simulation started at the beginning of 1998, and thus 1998.00 was entered as the time-offset value. Simulation time values in days were converted to values in years by specifying years as the type of output. Summing the time offset and the converted simulation time gives the simulation time in decimal years. HYDFMT has the capability of reporting time in formats involving separate values of years, months, and days.

```
 Enter name of file with unformatted hydrograph data:
testt.sav
 Enter root-name of file for formatted output (ex. hydro):
hydro
 DO YOU WANT DECIMAL OR CALENDAR (DATE) TIME FORMAT(D/C):
D
 ENTER UNIT DECIMAL TIME AS YEARS, DAYS, OR MINUTES (Y,D,M)FOR OUTPUT:
Y
 ENTER INITIAL DECIMAL TIME OF TRANSIENT SIMULATION (ex. 1891.00):
1998.00
 DO YOU IGNORE LEAP DAYS,
 DISTRIBUTE THEM UNIFORMLY OVER FOUR YEARS, OR
 DO YOU WANT THE LEAP DAY INSERTED INTO THE LEAP YEAR?
 ENTER A CHOICE FOR IGNORE, UNIFORM, OR LEAP (I/U/L):
I
 DO YOU WANT ALL OR SELECTED SITES(A/S):
A
```

Figure 4. Example session of *HYDMOD* post-processor program HYDFMT. Program prompts are given in normal text and user responses are given in bold text.

HYDFMT SOURCE CODE

The following HYDFMT program source code can be compiled using a standard Fortran 77 compiler. For compatibility with unformatted *MODFLOW* output, HYDFMT should be compiled with the same compiler that was used to compile the *MODFLOW* program.

```
CC
      PROGRAM HYDFMT
CC
C----- Reads unformatted data from the HYDMOD package and writes out
C----- columns as formatted data to selected files by topic. The files
C----- have extensions that represent the type of data they contain.
C----- The three-letter extensions are as follows:
C-----      TYPE OF DATA                                    OUTPUT FILE
C-----                                                      EXTENSION
C----- (1)  ground-water levels (head)                          .gwh
C----- (2)  ground-water level decline (drawdown)               .gwd
C----- (3)  ground-water preconsolidation level (critical head)  .crh
C----- (4)  layer-specific compaction (compaction)              .cmp
C----- (5)  sum of all layer compaction (subsidence)            .sub
C----- (6)  surface-water streamflow water level (stage)       .sth
C----- (7)  surface-water streamflow into a reach (inflow)     .inf
C----- (8)  surface-water streamflow out of a reach (inflow)   .otf
C----- (9)  flow between ground-water & surface-water (leakage) .gsf
C-----
C----- R.T. Hanson, USGS-WRD, San Diego, California, rthanson@usgs.gov
C----- S.A. Leake, USGS-WRD, Tucson, Arizona, saleake@usgs.gov
```

```
C-----
C----- VERSION 0100 29JULY1998 HYDFMT
C=======================================================================
      CHARACTER FN1*80,FN2*80,FMT1*80,FN3*80,WELLID*20
      CHARACTER TIMTYP*1,ELPTYP*1,arr*2,timlbl*4,fmtout*27,fmtoute*17
      character fmtoutd*20,blank*80,SITTYP*1,LEAPYR*1/'L'/
      character WELLIDH*20,WELLIDD*20,WELLIDCH*20,
     & WELLIDC*20,WELLIDS*20,WELLIDFI*20,
     & WELLIDHF*20,WELLIDFO*20,WELLIDFA*20
      DIMENSION JCOL(5000),Z(5000),zh(5000),id(5000),
     & zhd(5000),zhph(5000),zhc(5000),zhs(5000),
     & zfi(5000),zfo(5000),zhf(5000),zfa(5000)
     DIMENSION WELLID(5000),WELLIDH(5000),
     & WELLIDD(5000),WELLIDCH(5000),WELLIDC(5000),
     & WELLIDS(5000),WELLIDFI(5000), WELLIDHF(5000),
     & WELLIDFO(5000),WELLIDFA(5000)
      INTEGER DAYS,IDATE,nu(9),icnt(9)
      INTEGER*4 ISTRNG
      logical*4 FIRST/.TRUE./,GEN/.TRUE./
CC
      COMMON /T/YR,LEAPYR
cc
       DATA (wellid(i),i=1,5000)/5000*' ---------- '/,
     &(wellidh(i),i=1,5000)/5000*' ---------- '/,
     &(wellidd(i),i=1,5000)/5000*' ---------- '/,
     &(wellidch(i),i=1,5000)/5000*' ---------- '/,
     &(wellidc(i),i=1,5000)/5000*' ---------- '/,
     &(wellids(i),i=1,5000)/5000*' ---------- '/,
     &(wellidfi(i),i=1,5000)/5000*' ---------- '/,
     &(wellidfo(i),i=1,5000)/5000*' ---------- '/,
     &(wellidfa(i),i=1,5000)/5000*' ---------- '/,
     &(wellidhf(i),i=1,5000)/5000*' ---------- '/,
     &(nu(i),i=1,9)/61,62,63,64,65,66,67,68,69/,
     &(icnt(i),i=1,9)/9*0/
cc
       DATA blank/'
     &                    '/
C
      inunit=60
      fmt1='                                        '
      fmtout='(i4,1x,a4,12x,5000(1x,a20))'
      fmtoute='(f10.3,5000f21.2)'
      fmtoutd='(1x,i8,3x,5000f21.2)'
      TIMTYP='D'
      ELPTYP='Y'
      NDATE=1
      START=0.0
      numhh=0
      numhd=0
      numch=0
      numhc=0
      numhs=0
      numhfi=0
      numhfo=0
      numhfa=0
      numhhf=0
      LEN=0
      ISTRNG=80
      FN1=blank
      FN2=blank
```

```
      FN3=blank
      FMT1=blank
CC
C----- Read file names
    1 WRITE(*,*) 'Enter name of file with unformatted hydrograph data:'
      READ(*,'(A)') FN1
      IF(FN1.EQ.'        ') GO TO 100
      OPEN(UNIT=inunit,FILE=FN1,FORM='UNFORMATTED',ACCESS='SEQUENTIAL')
C----- Read first unformatted header record with number of hydrographs saved
c       and time units used in the model.
      READ(inunit) NUMH,ITMUNI
C----- Read second unformatted header record with labels of hydrographs saved
      READ(inunit) timlbl,(wellid(n)(1:20),n=1,numh)
      WRITE(*,*) 'Enter root-name of file for formatted output',
     &' (ex. hydro):'
      READ(*,'(A)') FN2
      CALL LNOTE(FN2,ISTRNG,LEN)
      if(LEN.gt.76)LEN=76
CC
      WRITE(*,*)'DO YOU WANT DECIMAL OR CALENDAR(DATE) ',
     &'TIME FORMAT(D/C):'
      READ(*,*)TIMTYP
      IF(TIMTYP.EQ.'C'.or.TIMTYP.EQ.'c')THEN
       WRITE(*,*)'ENTER INITIAL DATE OF TRANSIENT SIMULATION ',
     &'(YYYYMMDD ex. 19920827):'
       READ(*,*)ISTRT
       ISTT=(ISTRT/1000000)
       IST=ISTT*1000000
       ISTART=ISTRT-IST
       IDAYS=DAYS(ISTART)
      ELSE IF(TIMTYP.EQ.'D'.or.TIMTYP.EQ.'d')THEN
       WRITE(*,*)'ENTER UNIT DECIMAL TIME AS YEARS, DAYS, OR MINUTES',
     &' (Y,D,M) FOR OUTPUT:'
       READ(*,*)ELPTYP
       WRITE(*,*)'ENTER INITIAL DECIMAL TIME OF TRANSIENT',
     &' SIMULATION (ex. 1891.00):'
       READ(*,*)START
      ENDIF
      WRITE(*,*)'DO YOU IGNORE LEAP DAYS,'
      WRITE(*,*)'DISTRIBUTE THEM UNIFORMLY OVER FOUR YEARS, OR'
      WRITE(*,*)'DO YOU WANT THE LEAP DAY INSERTED INTO THE LEAP YEAR?'
      WRITE(*,*)'ENTER A CHOICE FOR IGNORE, UNIFORM, OR LEAP (I/U/L):'
      READ(*,*)LEAPYR
       if(timtyp.eq.'D'.or.timtyp.eq.'d')then
        FMT1(1:17)=fmtoute
       elseif(timtyp.eq.'C'.or.timtyp.eq.'c')then
        FMT1(1:20)=fmtoutd
       endif
       do 22 ii=1,numh
       CALL LEFTJ(wellid(ii))
          read(wellid(ii)(1:2),'(a2)')arr
       if(arr.eq.'HD')then
        id(ii)=1
        numhh=numhh+1
        wellidh(numhh)=wellid(ii)
       elseif(arr.eq.'DD')then
        id(ii)=2
        numhd=numhd+1
        wellidd(numhd)=wellid(ii)
       elseif(arr.eq.'HC')then
```

```
        id(ii)=3
        numch=numch+1
        wellidch(numch)=wellid(ii)
       elseif(arr.eq.'CP')then
        id(ii)=4
        numhc=numhc+1
        wellidc(numhc)=wellid(ii)
       elseif(arr.eq.'SB')then
        id(ii)=5
        numhs=numhs+1
        wellids(numhs)=wellid(ii)
       elseif(arr.eq.'ST')then
        id(ii)=6
        numhhf=numhhf+1
        wellidhf(numhhf)=wellid(ii)
       elseif(arr.eq.'SI')then
        id(ii)=7
        numhfi=numhfi+1
        wellidfi(numhfi)=wellid(ii)
       elseif(arr.eq.'SO')then
        id(ii)=8
        numhfo=numhfo+1
        wellidfo(numhfo)=wellid(ii)
       elseif(arr.eq.'SA')then
        id(ii)=9
        numhfa=numhfa+1
        wellidfa(numhfa)=wellid(ii)
       endif
  22   continue
cc
      if(numhh.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.gwh'
       OPEN(UNIT=nu(1),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(1),fmt=fmtout)numhh,timlbl,
     & (wellidh(n)(1:20),n=1,numhh)
      endif
      if(numhd.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.gwd'
       OPEN(UNIT=nu(2),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(2),fmt=fmtout)numhd,timlbl,
     & (wellidd(n)(1:20),n=1,numhd)
      endif
      if(numch.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.crh'
       OPEN(UNIT=nu(3),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(3),fmt=fmtout)numch,timlbl,
     & (wellidch(n)(1:20),n=1,numch)
      endif
      if(numhc.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.cmp'
       OPEN(UNIT=nu(4),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(4),fmt=fmtout)numhc,timlbl,
     & (wellidc(n)(1:20),n=1,numhc)
      endif
      if(numhs.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.sub'
       OPEN(UNIT=nu(5),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(5),fmt=fmtout)numhs,timlbl,
     & (wellids(n)(1:20),n=1,numhs)
      endif
```

```
      if(numhhf.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.sth'
       OPEN(UNIT=nu(6),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(6),fmt=fmtout)numhhf,timlbl,
     & (wellidhf(n)(1:20),n=1,numhhf)
       endif
       if(numhfi.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.inf'
       OPEN(UNIT=nu(7),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(7),fmt=fmtout)numhfi,timlbl,
     & (wellidfi(n)(1:20),n=1,numhfi)
       endif
       if(numhfo.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.otf'
       OPEN(UNIT=nu(8),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(8),fmt=fmtout)numhfo,timlbl,
     & (wellidfo(n)(1:20),n=1,numhfo)
       endif
       if(numhfa.gt.0)then
       write(FN2(LEN+1:LEN+4),'(A)')'.gsf'
       OPEN(UNIT=nu(9),FILE=FN2,FORM='FORMATTED',ACCESS='SEQUENTIAL')
       write(nu(9),fmt=fmtout)numhfa,timlbl,
     & (wellidfa(n)(1:20),n=1,numhfa)
       endif
cc
C----- Read the number of columns to be written to output file
C----- The number of columns does not include the simulation time which is
C----- always the first column in the formatted file.
      WRITE(*,*)'DO YOU WANT ALL OR SELECTED SITES(A/S):'
      READ(*,*)SITTYP
      IF(SITTYP.EQ.'A'.OR.SITTYP.EQ.'a')then
       GEN=.TRUE.
       NC=numh
       DO 21 IK=1,numh
  21    JCOL(IK)=IK
      ELSEIF(SITTYP.EQ.'S'.OR.SITTYP.EQ.'s')then
   5   WRITE(*,*) 'Enter number of columns to be read:'
       READ(*,*) NC
       WRITE(*,10) NC
       GEN=.FALSE.
C----- Read the column numbers, free format, any order
   10  FORMAT('Enter ',I3,' column numbers:')
       READ(*,*) (JCOL(N),N=1,NC)
C----- Read format for output of data, free format, no quotes.  Example:
C-----      (F10.2,4F10.3)
C----- This example writes simulation time with F10.3 and thirty columns of
C----- observations with F10.2
       WRITE(*,*)'Enter format for output (ex (F10.3,30F10.2) or (*):'
       READ(*,'(A)') FMT1
      DO 25 N=1,NC
      IF(JCOL(N).GT.NUMH) THEN
       WRITE(*,20) JCOL(N),NUMH
   20  FORMAT(' Column number',I3,' is greater than maximum possible of',
     1        I3,/,' please reenter column numbers...')
       GO TO 5
      ENDIF
   25 CONTINUE
      ENDIF
C----- Read the unformatted and Write the formatted results
   30 READ(inunit,END=50) TIME,(Z(N),N=1,NUMH)
```

```
C----- Convert model elapsed time into calendar time
      CALL MODTIME(ITIME,TIME,ITMUNI,IDAYS,IDATE,ISTART,TIMTYP,
     -ELPTYP,FIRST,START)
      FIRST=.FALSE.
      IF(TIMTYP.EQ.'C'.or.TIMTYP.EQ.'c')IDATE=IDATE+IST
cc
       do 89 ii=1,NC
       if(id(ii).gt.0)icnt(id(ii))=icnt(id(ii))+1
       if(id(ii).eq.1)then
        zh(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.2)then
        zhd(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.3)then
        zhph(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.4)then
        zhc(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.5)then
        zhs(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.6)then
        zhf(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.7)then
        zfi(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.8)then
        zfo(icnt(id(ii)))=Z(JCOL(ii))
       elseif(id(ii).eq.9)then
        zfa(icnt(id(ii)))=Z(JCOL(ii))
       endif
  89  continue
cc
      IF(FMT1(1:3).EQ.'(*)')THEN
       IYEAR=IDATE/10000
       IMON=(IDATE-(IYEAR*10000))/100
       IDAY=(IDATE-(IYEAR*10000)-(IMON*100))
       IF(IYEAR.LE.99)IYEAR=1900+IYEAR
       IF(IDATE.LT.ISTART)IYEAR=2000+IYEAR
       IDATE=IMON*1000000+IDAY*10000+IYEAR
      if(numhh.gt.0)then
       DO 91 N=1,numhh
 91    WRITE(nu(1),*)WELLIDH(N),',',IDATE,',',ZH(N)
      endif
      if(numhd.gt.0)then
       DO 92 N=1,numhd
 92    WRITE(nu(2),*)WELLIDD(N),',',IDATE,',',ZHD(N)
      endif
      if(numch.gt.0)then
       DO 93 N=1,numch
 93    WRITE(nu(3),*)WELLIDCH(N),',',IDATE,',',ZHPH(N)
      endif
      if(numhc.gt.0)then
       DO 94 N=1,numhc
 94    WRITE(nu(4),*)WELLIDC(N),',',IDATE,',',ZHC(N)
      endif
      if(numhs.gt.0)then
       DO 95 N=1,numhs
 95    WRITE(nu(4),*)WELLIDS(N),',',IDATE,',',ZHS(N)
      endif
      if(numhhf.gt.0)then
       DO 96 N=1,numhhf
 96    WRITE(nu(6),*)WELLIDHF(N),',',IDATE,',',ZHF(N)
      endif
```

```
      if(numhfi.gt.0)then
       DO 97 N=1,numhfi
 97    WRITE(nu(7),*)WELLIDFI(N),',',IDATE,',',ZFI(N)
      endif
      if(numhfo.gt.0)then
       DO 98 N=1,numhfo
 98    WRITE(nu(8),*)WELLIDFO(N),',',IDATE,',',ZFO(N)
      endif
      if(numhfa.gt.0)then
       DO 99 N=1,numhfa
 99    WRITE(nu(9),*)WELLIDFA(N),',',IDATE,',',ZFA(N)
      endif
      ELSE IF (FMT1(1:3).NE.'(*)'.and.(TIMTYP.EQ.'C'.or.
     &TIMTYP.EQ.'c'))THEN
       if(numhh.gt.0)WRITE(nu(1),FMT1) IDATE,(ZH(N),N=1,numhh)
       if(numhd.gt.0)WRITE(nu(2),FMT1) IDATE,(ZHD(N),N=1,numhd)
       if(numch.gt.0)WRITE(nu(3),FMT1) IDATE,(ZHPH(N),N=1,numch)
       if(numhc.gt.0)WRITE(nu(4),FMT1) IDATE,(ZHC(N),N=1,numhc)
       if(numhs.gt.0)WRITE(nu(5),FMT1) IDATE,(ZHS(N),N=1,numhs)
       if(numhhf.gt.0)WRITE(nu(6),FMT1) IDATE,(ZHF(N),N=1,numhhf)
       if(numhfi.gt.0)WRITE(nu(7),FMT1) IDATE,(ZFI(N),N=1,numhfi)
       if(numhfo.gt.0)WRITE(nu(8),FMT1) IDATE,(ZFO(N),N=1,numhfo)
       if(numhfa.gt.0)WRITE(nu(9),FMT1) IDATE,(ZFA(N),N=1,numhfa)
      ELSE IF (FMT1(1:3).NE.'(*)'.and.(TIMTYP.EQ.'D'.or.
     &TIMTYP.EQ.'d'))THEN
       if(numhh.gt.0)WRITE(nu(1),FMT1) TIME,(ZH(N),N=1,numhh)
       if(numhd.gt.0)WRITE(nu(2),FMT1) TIME,(ZHD(N),N=1,numhd)
       if(numch.gt.0)WRITE(nu(3),FMT1) TIME,(ZHPH(N),N=1,numch)
       if(numhc.gt.0)WRITE(nu(4),FMT1) TIME,(ZHC(N),N=1,numhc)
       if(numhs.gt.0)WRITE(nu(5),FMT1) TIME,(ZHS(N),N=1,numhs)
       if(numhhf.gt.0)WRITE(nu(6),FMT1) TIME,(ZHF(N),N=1,numhhf)
       if(numhfi.gt.0)WRITE(nu(7),FMT1) TIME,(ZFI(N),N=1,numhfi)
       if(numhfo.gt.0)WRITE(nu(8),FMT1) TIME,(ZFO(N),N=1,numhfo)
       if(numhfa.gt.0)WRITE(nu(9),FMT1) TIME,(ZFA(N),N=1,numhfa)
      ENDIF
      do 51 ix=1,9
   51 icnt(ix)=0
      GO TO 30
   50 CLOSE(inunit)
      if(numhh.gt.0)CLOSE(nu(1))
      if(numhd.gt.0)CLOSE(nu(2))
      if(numch.gt.0)CLOSE(nu(3))
      if(numhc.gt.0)CLOSE(nu(4))
      if(numhs.gt.0)CLOSE(nu(5))
      if(numhhf.gt.0)CLOSE(nu(6))
      if(numhfi.gt.0)CLOSE(nu(7))
      if(numhfo.gt.0)CLOSE(nu(8))
      if(numhfa.gt.0)CLOSE(nu(9))
  100 STOP
      END
C*************************************************************************
      SUBROUTINE MODTIME(ITIME,TOTIM,ITMUNI,IDAYS,IDATE,ISTART,
     -TIMTYP,ELPTYP,FIRST,START)
CC
C-----
C----- VERSION 0100 29JULY1998 MODTIME
C     ***********************************************************************
C     CONVERT ELAPSED MODEL TIME TO DECIMAL OR CLAENDAR DATES FOR HYDROGRAPHS
C     ***********************************************************************
C
```

```
C     SPECIFICATIONS:
C     --------------------------------------------------------------------
      CHARACTER TIMTYP*1,ELPTYP*1,LEAPYR*1
      INTEGER DATE,DAYS
      LOGICAL*4 FIRST
CC
      COMMON /T/YR,LEAPYR
CC

      FYY=0.
      IYY=0
      IF(FIRST)REMTIM=0.0
      IF(LEAPYR.EQ.'U')THEN
       YR=365.25
      ELSE
       YR=365.
      ENDIF
CC
      IF(ITMUNI.EQ.0)THEN
      WRITE(*,*)'TIME UNIT OF MODEL DATA IS UNDEFINED CHECK',
     -'BASIC PACKAGE DATA-->ITMUNI'
      STOP
      ENDIF
CC
CC    SET MULTIPLIER BASED ON MODEL TIME, FOR DATE FORMAT CONVERSION
CC        TO ELAPSED DAYS
CC
      IF(TIMTYP.EQ.'C'.or.TIMTYP.EQ.'c')THEN
       IF(ITMUNI.EQ.1)THEN
        DELTIM=86400.
       ELSE IF(ITMUNI.EQ.2)THEN
        DELTIM=1440.
       ELSE IF(ITMUNI.EQ.3)THEN
        DELTIM=24.
       ELSE IF(ITMUNI.EQ.4)THEN
        DELTIM=1.
       ELSE IF(ITMUNI.EQ.5)THEN
        DELTIM=1./YR
       ENDIF
       ITIME=TOTIM/DELTIM
       REMTIM=(TOTIM/DELTIM)-INT(TOTIM/DELTIM)+REMTIM
       IF((REMTIM-1.).GT.0.)THEN
        REMTIM=REMTIM-1.
        ITIME=ITIME+1
       ENDIF
       ITIME=ITIME+IDAYS
       ITIME2=ITIME
       IDATE=DATE(ITIME)
       IF(ITMUNI.LT.5)THEN
        YY=(IDATE/10000)-(ISTART/10000)
        IF(LEAPYR.EQ.'L')THEN
         IF(MOD(YY,4.).EQ.0.)ITIME2=ITIME2+INT(YY/4.)+1
         IF(MOD(YY,4.).NE.0.)ITIME2=ITIME2+INT(YY/4.)
        ENDIF
        IDATE=DATE(ITIME2)
       ENDIF
CC
CC    SET MULTIPLIER BASED ON MODEL TIME FOR DECIMAL TIME FLAG ELPTYP
CC
      ELSE IF(TIMTYP.EQ.'D'.or.TIMTYP.EQ.'d')THEN
```

```
      IF(ELPTYP.EQ.'M'.or.ELPTYP.EQ.'m')THEN
       IF(ITMUNI.EQ.1)THEN
        DELTIM=60.
       ELSE IF(ITMUNI.EQ.2)THEN
        DELTIM=1.
       ELSE IF(ITMUNI.EQ.3)THEN
        DELTIM=1/60.
       ELSE IF(ITMUNI.EQ.4)THEN
        DELTIM=1/1440.
       ELSE IF(ITMUNI.EQ.5)THEN
        DELTIM=1./525600.
       ENDIF
      ELSE IF(ELPTYP.EQ.'D'.or.ELPTYP.EQ.'d')THEN
       IF(ITMUNI.EQ.1)THEN
        DELTIM=86400.
       ELSE IF(ITMUNI.EQ.2)THEN
        DELTIM=1440.
       ELSE IF(ITMUNI.EQ.3)THEN
        DELTIM=24.
       ELSE IF(ITMUNI.EQ.4)THEN
        DELTIM=1.
       ELSE IF(ITMUNI.EQ.5)THEN
        DELTIM=1./YR
       ENDIF
      ELSE IF(ELPTYP.EQ.'Y'.or.ELPTYP.EQ.'y')THEN
       IF(ITMUNI.EQ.1)THEN
        DELTIM=31536000.
       ELSE IF(ITMUNI.EQ.2)THEN
        DELTIM=525600.
       ELSE IF(ITMUNI.EQ.3)THEN
        DELTIM=8760.
       ELSE IF(ITMUNI.EQ.4)THEN
        DELTIM=YR
       ELSE IF(ITMUNI.EQ.5)THEN
        DELTIM=1.
       ENDIF
      TOTIM=(TOTIM/DELTIM)+START
      GOTO 99
      ENDIF
     ISTY=(ISTART/10000)*10000
     IYY=DAYS(ISTART)-DAYS(ISTY)-1
     FYY=FLOAT(IYY)/YR
     YY=START+(ISTART/10000)+FYY
     TOTIM=TOTIM/DELTIM
     ENDIF
  99  RETURN
     END
C
C*******************************************************************************
CC
     INTEGER FUNCTION DAYS(DAT)
C-----
C----- VERSION 0100 29JULY1998 DAYS
C     ***********************************************************************
CC     FOR DATE GIVEN AS AN INTEGER YYMMDD, THIS FUNCTION RETURNS
CC     THE NUMBER OF DAYS SINCE DECEMBER 31, 1899.
C     ***********************************************************************
C
C     SPECIFICATIONS:
C     ----------------------------------------------------------------------
```

```
CC
      INTEGER DAT,YEAR,DAY,DAYM(12)
      CHARACTER*1 LEAPYR
CC
      COMMON /T/YR,LEAPYR
CC
      DATA DAYM/0,31,59,90,120,151,181,212,243,273,304,334/
CC
      YEAR=DAT/10000
      M=DAT-YEAR*10000
      MONTH=M/100
      DAY=M-MONTH*100
      IF(MONTH.LT.1)MONTH=1
      IF(DAY.LT.1)DAY=1
      IF(LEAPYR.EQ.'U')THEN
       YR=365.25
       DAYS=YEAR*YR
      ELSEIF(LEAPYR.EQ.'L')THEN
       YR=365.
       DAYS=YEAR*YR+(YEAR+3)/4
      ELSE
       YR=365.
       DAYS=YEAR*YR
      ENDIF
      DAYS=DAYS+DAYM(MONTH)+DAY
      IF(MONTH.LE.2)RETURN
      IF(MOD(YEAR,4).NE.0) RETURN
      IF(LEAPYR.EQ.'L')DAYS=DAYS+1
      RETURN
      END
C*******************************************************************************
      INTEGER FUNCTION DATE(DAYS)
C-----
C----- VERSION 0100 29JULY1998 DATE
C     ************************************************************************
CC     FOR VALUE OF DAYS SINCE DECEMBER 31, 1899 FUNCTION RETURNS THE
CC     DATE AS AN INTEGER IN THE FORM YYMMDD
C     ************************************************************************
C
C     SPECIFICATIONS:
C     ---------------------------------------------------------------------
      INTEGER DAYS,DAYM(12,2),YY,MM,DD,DY,NL
CC
      DATA ((DAYM(I,J),I=1,12),J=1,2)
     &/0,31,59,90,120,151,181,212,243,273,304,334,
     & 0,31,60,91,121,152,182,213,244,274,305,335/
CC
      L=1
      NL=0
      IF(DAYS.GE.61) NL=(DAYS-61)/1461+1
      YY=(DAYS-NL-1)/365
      IF(MOD(YY,4).EQ.0) L=2
      DY=DAYS-365*YY-YY/4-(2-L)
      MM=0
   10 MM=MM+1
      IF(MM.EQ.12) GO TO 20
      IF(DY.GT.DAYM(MM+1,L)) GO TO 10
   20 DD=DY-DAYM(MM,L)
      DATE=10000*YY+100*MM+DD
      RETURN
```

```
      END
CC*****************************************************************************
      SUBROUTINE LNOTE(STRING,DIMS,LEN)
C-----
C----- VERSION 0100 29JULY1998 LNOTE
C     ********************************************************************
CC     RETURNS THE LEFT-JUSTIFIED, NONBLANK LENGTH
CC     FOR A CHARACTER STRING
C     ********************************************************************
C
C     SPECIFICATIONS:
C     --------------------------------------------------------------------
      INTEGER*4 DIMS,LEN
      CHARACTER STRING*80
      DO 10 N=DIMS,1,-1
      IF(STRING(N:N).NE.' ')GO TO 20
   10 CONTINUE
      LEN=0
      RETURN
   20 LEN=N
      RETURN
      END
CC*****************************************************************************
      SUBROUTINE LEFTJ(STRING)
C-----
C----- VERSION 0100 29JULY1998 LEFTJ
C     ********************************************************************
CC     RETURNS THE LEFT-JUSTIFIED, NONBLANK
CC     PORTION OF A CHARACTER STRING
C     ********************************************************************
C
C     SPECIFICATIONS:
C     --------------------------------------------------------------------
      CHARACTER STRING*20
      DO 10 I=1,20
      IF(STRING(I:I).NE.' ') GO TO 20
   10 CONTINUE
      RETURN
   20 STRING=STRING(I:20)
      RETURN
      END
CC*************************************
```

The Hydrograph program (*HYDMOD*) documented in this report generates hydrograph information for specific locations during simulations using *MODFLOW*. *HYDMOD* is a separate program within *MODFLOW* and requires its own input file that specifies the data types and locations for saving time-series data from the simulation. Program *HYDPOST*, in contrast, is a simple *MODFLOW* post-processor program that allows extraction of time-series of model-computed head, drawdown, and other variables for selected model cells from unformatted output directly from *MODFLOW*. Thus, *HYDPOST* has been included here as a simple alternative way of extracting hydrograph information from unformatted arrays of head or other data that were specified for output at user-specified time steps in the *MODFLOW* output-control input file and recorded during a *MODFLOW* simulation. Unlike *HYDMOD*, the use of *HYDPOST* does not require any new packages, modification of *MODFLOW*, or additional *MODFLOW* input files.

USE OF THE *HYDPOST* PROGRAM

Prior to using *HYDPOST*, a user must have run *MODFLOW* and saved head, drawdown, or other information from *MODFLOW* in an unformatted file at all or at selected time steps. Information that can be read by *HYDPOST* is limited to arrays written to files using the utility module ULASAV in *MODFLOW*. These arrays include head and drawdown in the Block-Centered-Flow Package; and compaction, subsidence, and critical head in the Interbed-Storage Package (Leake and Prudic, 1991). Output from *HYDPOST* includes a file containing time and a series of hydrograph values for model nodes of interest. Additionally, information on unformatted arrays contained in the *MODFLOW* output file is written to a file named MESSAGE.OUT.

Input to *HYDPOST* is interactive. Users must specify items including file names, number and grid locations of hydrograph points, and parameters for transforming model simulation time into another time system. The *HYDPOST* information prompts and example user responses are shown in figure 5.

For each execution of the *HYDPOST* program, all hydrograph points must be in a single model layer. If desired hydrograph points are in multiple model layers, *HYDPOST* must be run once for each model layer containing one or more hydrograph points.

The time stored with each unformatted array recorded by *MODFLOW* is the elapsed time since the start of the simulation. The units of time recorded are specified by the ITMUNI variable in the *MODFLOW* Basic Package input (McDonald and Harbaugh, 1988). To convert simulation time to real-world time in the *HYDPOST* output, users can enter a time-offset and a time-conversion factor. In the example shown in figure 5, model simulation time is in seconds, but time values in calendar years, including fractions, are desired for hydrograph output. The simulation started at the beginning of 1965, and thus 1965.0 was entered as the time-offset value. Simulation time values in seconds were converted to values in years by entering the factor of 3.1688×10^{-8} (3.1688E-08), which is the fraction of an average year in 1 second. Summing the time offset and the converted simulation time gives the simulation time in calendar years. Unlike *HYDMOD* postprocessing with HYDFMT, *HYDPOST* does not have the capability of recording time in formats involving separate values of years, months, and days.

```
 Enter name of BAS Package input file:
bas.pckg
 Enter name of file with unformatted arrays from MODFLOW:
bm.hd.temp
 Name of unformatted array (i.e. HEAD, DRAWDOWN):
HEAD
 Enter name of file for formatted array:
hyd.out
 Enter number of hydrograph points to save:
2
 All points must be in the same model layer. Enter layer number:
1
 Enter 2 pairs of row and column locations:
33,40
60,51
 Do you want to transform simulation time to real-
 world time using the relation:
 (real time) = (offset) + (factor)*(simulation time) (Y/N)?
Y
 Enter offset:
1965.0
 Enter factor:
3.1688E-08
 Enter format for output, for example, (2F10.2):
(F10.3,2F10.2)
 Do you want a screen echo of arrays in file (Y/N)?
N
```

Figure 5. Example session of program *HYDPOST*. Program prompts are given in normal text and user responses are given in bold text.

The format for the output of time and hydrograph values must be entered interactively as a valid **FORTRAN** format. The format specifier must include an opening parenthesis, a format for the time, formats for each of the hydrograph points, and a closing parenthesis. In the example in figure 5, the format specifier is **(F10.3,2F10.2)**. The format **F10.3** for the time value results in a floating-point field width of 10 spaces with three digits to the right of the decimal point. The format **2F10.2** results in a field width of 10 spaces for each of the two head values with two digits to the right of the decimal point. For more details of format specifiers, refer to any standard **FORTRAN** manual.

HYDPOST SOURCE CODE

The following HYDPOST program source code can be compiled using a standard FORTRAN compiler. For compatibility with MODFLOW output, HYDPOST should be compiled with the same compiler that was used to compile the MODFLOW program.

```
C.....   Program HYDPOST--Hydrograph post processor
C.....   This program is a post-processor for MODFLOW that reads a file
C.....   containing unformatted arrays written with module ULASAV and
C.....   writes simulation time and up to 20 head values (or other values)
C.....   to a file that can be read into a graphing program.
C.....
C.....     HYDPOST Version 1.0        February 12, 1998
C.....
C-----------------------------------------------------------------------
      CHARACTER NAME*10,FN1*40,FN3*40,TEXT*16,ANS*1,STR10*10,FMT1*40
      CHARACTER DUMMY*1
      LOGICAL FOUND,ECHO,EXISTS,EOF,TTRAN
C-----------------------------------------------------------------------
C..... Dimension array Zn at least as big as the number of model cells
C..... in one layer of the grid. Arrays JC, IR, and IJ are dimensioned
C..... for a maximum of 20 hydrograph points
C-----------------------------------------------------------------------
      DIMENSION Zn(40000),JC(20),IR(20),IJ(20)
      DATA NCL,NRW/13,1/
      OPEN(UNIT=63,FILE='MESSAGE.OUT')
C----- Read file name of BAS Package input, determine number of
C----- layers, rows, and columns
      WRITE(*,*)' Enter name of BAS Package input file:'
      READ(*,'(A)') FN1
      OPEN(UNIT=79,FILE=FN1)
      READ(79,'(A1)') DUMMY
      READ(79,'(A1)') DUMMY
      READ(79,*) NLAY,NROW,NCOL
      CLOSE(79)
C----- Read name of file containing MODFLOW unformatted arrays of
C----- interest. Open file.
      WRITE(*,*)' Enter name of file with unformatted arrays',
     $ ' from MODFLOW: '
      READ(*,'(A)') FN1
      OPEN(UNIT=79,FILE=FN1,FORM='UNFORMATTED')
C----- Read name of array from which hydrograph values will be saved
      WRITE(*,*) ' Name of unformatted array (i.e. HEAD, DRAWDOWN):'
      READ(*,'(A)') NAME
C----- Read name of file for saving hydrograph output. Open file.
      WRITE(*,*) ' Enter name of file for formatted array:'
  101 READ(*,'(A)') FN3
      INQUIRE(FILE=FN3,EXIST=EXISTS)
      IF(.NOT.EXISTS) THEN
       OPEN(UNIT=64,FILE=FN3)
      ELSE
       WRITE(*,*) ' Output file already exists. overwrite? (Y/N):'
       READ(*,'(A)') ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
        OPEN(UNIT=64,FILE=FN3)
       ELSE
        WRITE(*,*) ' Please enter a different file name:'
        GO TO 101
```

```
       ENDIF
      ENDIF
C----- Read number and grid locations of hydrograph points.
      WRITE(*,*) ' Enter number of hydrograph points to save:'
      READ(*,*) NPTS
      WRITE(*,*) ' All points must be in the same model layer. ',
     $ ' Enter layer number: '
      READ(*,*) KLAY
      WRITE(*,10) NPTS
   10 FORMAT(' Enter',I3,' pairs of row and column locations: ')
      READ(*,*) (IR(K),JC(K),K=1,NPTS)
      DO 12 K=1,NPTS
      IJ(K)=(IR(K)-1)*NCOL+JC(K)
   12 CONTINUE
C----- Read time-transformation parameters, if option is desired.
      WRITE(*,*) ' Do you want to transform simulation time to real-'
      WRITE(*,*) ' world time using the relation:'
      WRITE(*,*) '   (real time) = (offset) + ',
     $ ' (factor)*(simulation time)         (Y/N)?'
      READ(*,'(A)') ANS
      TTRAN=.FALSE.
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
       TTRAN=.TRUE.
      WRITE(*,*) ' Enter offset:'
      READ(*,*) TOFFST
      WRITE(*,*) ' Enter factor:'
      READ(*,*) TFACT
      ENDIF
C----- Enter format for hydrograph output.
      WRITE(*,*) ' Enter format for output, for example, (2F10.2):'
      READ(*,'(A)') FMT1
C----- Read each unformatted array with the appropriate values. If user
C----- requests, list each unformatted array found in the file.
      WRITE(*,*) ' Do you want a screen echo of arrays in file (Y/N)?'
      READ(*,'(A)') ANS
      ECHO=.FALSE.
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
       ECHO=.TRUE.
      ENDIF
      IF(ECHO) THEN
       WRITE(*,*) '        The following arrays were found--'
       WRITE(*,*) '        STEP     PERIOD    LAYER      NAME'
      ENDIF
       WRITE(63,*) '        The following arrays were found--'
       WRITE(63,*) '        STEP     PERIOD    LAYER      NAME'
  200 CALL ZREAD2(Zn,KLAY,NAME,KSTP,KPER,PERTIM,TOTIM,
     $           TEXT,ILAY,NCOL,NROW,FOUND,ECHO,EOF,79)
      IF(EOF) GO TO 999
      IF(.NOT.FOUND) GO TO 200
C..... Write hydrograph record for the array currently in memory.
      IF(TTRAN) THEN
       WRITE(64,FMT1) TOFFST+TFACT*TOTIM,(Zn(IJ(K)),K=1,NPTS)
      ELSE
       WRITE(64,FMT1) TOTIM,(Zn(IJ(K)),K=1,NPTS)
      ENDIF
      GO TO 200
C..... All arrays ahve been read. End Program.
  999 STOP
      END
      SUBROUTINE ZREAD2(Z,ILAYER,N10,KSTP,KPER,PERTIM,
```

```
     $ TOTIM,TEXT,ILAY,NCL,NRW,FOUND,ECHO,EOF,IUNIT)
C...... Read Z values from an unformmated file into an array dimensioned
C.....  as NCL,NRW. Arrays for which the first 10 non-blank characters
C...... do not match the text string N10 will be skipped.
C......
C-----------------------------------------------------------------------
      CHARACTER TEXT(4)*4,N10*10,ANAME*16,NAME*20
      LOGICAL FOUND, ECHO,EOF
      DIMENSION Z(NCL,NRW)
C-----------------------------------------------------------------------
      EOF=.FALSE.
      FOUND=.FALSE.
       NAME=N10//'          '
C----- Make sure first character in string is non-blank.
       CALL LEFTJ20(NAME)
C----- read an unformatted header record and concatenate name into a
C----- single string.
   10  READ(IUNIT,END=8) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY
       ANAME=TEXT(1)//TEXT(2)//TEXT(3)//TEXT(4)
C----- Set starting point in string to be first non-blank character.
      DO 11 NS=1,16
      IF(ANAME(NS:NS).NE.' ') GO TO 12
   11 CONTINUE
   12 ANAME=ANAME(NS:16)
C----- Print information on unformatted array found, if requested.
      IF(ECHO) THEN
       WRITE(*,'(3I10,1X,A)') KSTP,KPER,ILAY,ANAME
      ENDIF
C----- Write information on array found to file.
       WRITE(63,'(3I10,1X,A)') KSTP,KPER,ILAY,ANAME
C----- Read unformatted array.
       READ(IUNIT) ((Z(J,I),J=1,NCOL),I=1,NROW)
       FOUND=.FALSE.
C----- If array does not have the correct name or layer, return with
C----- flag set indicating array is not of interest.
       IF(ANAME(1:10).NE.NAME) RETURN
       IF(ILAYER.NE.ILAY) RETURN
C----- If array has the correct name and layer, return with
C----- flag set indicating array is of interest.
       FOUND=.TRUE.
      RETURN
    8 EOF=.TRUE.
      RETURN
      END
      SUBROUTINE LEFTJ20(STRING)
C----- Left-justify non-blank characters in a string of 20 characters.
C-----------------------------------------------------------------------
      CHARACTER STRING*20
C-----------------------------------------------------------------------
C----- Find first non-blank character. Return if string is entirely
C----- blank.
      DO 10 I=1,20
      IF(STRING(I:I).NE.' ')GO TO 20
   10 CONTINUE
      RETURN
C----- Set starting point in string to be first non-blank character.
   20 STRING=STRING(I:20)
      RETURN
      END
```