Techniques of Water-Resources Investigations of the United States Geological Survey

Chapter A1

A MODULAR THREE-DIMENSIONAL FINITE-DIFFERENCE GROUND-WATER FLOW MODEL

By Michael G. McDonald and Arlen W. Harbaugh

This chapter supersedes U.S. Geological Survey Open-File Report 83-875

Book 6

MODELING TECHNIQUES
Module Documentation for the Basic Package

The Basic Package (BASl) consists of eight primary modules and five submodules. The modules are:

### Primary Modules

- **BAS1DF**: Defines and sets key model parameters.
- **BAS1AL**: Allocates space for data arrays used by the Basic Package.
- **BAS1RP**: Reads and prepares data for the Basic Package.
- **BAS1ST**: Reads timing information and initializes variables needed to calculate the length of time steps.
- **BAS1AD**: Calculates the length of time steps, accumulates elapsed time, and initializes heads at the beginning of each time step.
- **BAS1FM**: Clears accumulators RHS and HCOF.
- **BAS1OC**: Sets flags which indicate when data should be printed or recorded on disk.
- **BAS1OT**: Prints and records heads, drawdowns, and overall volumetric budget.

### Submodules

- **SBAS1D**: Calculates, writes, and records drawdown distribution.
- **SBAS1H**: Writes and records head distribution.
- **SBAS1I**: Initializes the Output Control System.
- **SBAS1T**: Prints a time summary.
- **SBAS1V**: Calculates and prints the overall volumetric budget.
Narrative for Module BAS1DF

The BAS1DF module defines and sets key model parameters. It does so in the following order:

1. Print the name of the program.
2. Read and print a heading.
3. Read the number of layers, rows, columns, stress periods, and units of time code ITMUNI. ITMUNI is a code which indicates the time units of model data. It does not affect model calculations but is used when printing the amount of elapsed time (see the input instructions for the codes).
4. Print the number of layers, rows, columns, and stress periods.
5. Select and print a message showing the time units.
6. Read and print the input unit numbers IUNIT for all major options. IUNIT is a 24-element table. Each entry has been assigned to a particular major option. The user specifies that a certain major option is to be used by putting a positive integer into the IUNIT entry corresponding to that major option. The integer is the unit number from which input to the major option will be read. If a major option is not going to be used, the corresponding IUNIT element is set equal to zero.
7. Initialize the total-elapsed time counter (TOTIM) and the storage-array counter (ISUM) and calculate the total number of cells.
8. RETURN.
ITMUNI is a code which indicates units of time used in the input data. This code is only used to print a table showing elapsed time in seconds, minutes, hours, days, and years. It is not used in formulating or solving the finite-difference equation.

IUNIT is a table that indicates which major options are to be used and the unit numbers from which input is to be read.

TOTIM is an accumulator in which total simulation time is stored. It is incremented at each time step.

ISUM is a location counter for the first unallocated space in the X array. It is incremented by each module in the Allocate Procedure.
SUBROUTINE BASDF(ISUMI, HEADING, NPER, ITMUNI, TOTIM, NCOL, NROW,
1         NLAY, NODES, INBAS, IOUT, IUNIT)

C-----VERSION 1513 12MAY1987 BASDF
C ******************************************************************
C DEFINE KEY MODEL PARAMETERS
C ******************************************************************
C
C SPECIFICATIONS:
C ------------------L-___________________________________L----------------
&------PRINT THE NAME OF THE PROGRAM.
WRITE(IOUT,1)
1 FORMAT(lhl,20x,'U. S. GEOLOGICAL SURVEY MODULAR',
1   'FINITE-DIFFERENCE GROUND-WATER MODEL')
C
C2------READ AND PRINT A HEADING.
READ(INBAS,2) HEADING
2 FORMAT(lhA4)
WRITE(IOUT,3) HEADING
3 FORMAT(lh32A4)
C
C3------READ NUMBER OF LAYERS, ROWS, COLUMNS, STRESS PERIODS AND
C3------UNITS OF TIME CODE.
READ(INBAS,4) NLAY, NROW, NCOL, NPER, ITMUNI
4 FORMAT(810)
C
C4------PRINT # OF LAYERS, ROWS, COLUMNS AND STRESS PERIODS.
WRITE(IOUT,5) NLAY, NROW, NCOL
5 FORMAT(lx, i4, ' LAYERS', liO, ' ROWS', liO, ' COLUMNS')
WRITE(IOUT,6) NPER
6 FORMAT(lx, i3, 'STRESS PERIOD(S) IN SIMULATION')
C
C5------SELECT AND PRINT A MESSAGE SHOWING TIME UNITS.
IF(ITMUNI.LT.0 .OR. ITMUNI.GT.5) ITMUNI=0
GO TO (10,20,30.40,50) ITMUNI
WRITE(IOUT,9)
9 FORMAT(lx,'MODEL TIME UNITS ARE UNDEFINED')
GO TO 100
10 WRITE(IOUT,11)
11 FORMAT(lx,'MODEL TIME UNIT IS SECONDS')
GO TO 100
20 WRITE(IOUT,21)
21 FORMAT(lx,'MODEL TIME UNIT IS MINUTES')
GO TO 100
30 WRITE(IOUT,31)
31 FORMAT(lx,'MODEL TIME UNIT IS HOURS')
GO TO 100
40 WRITE(IOUT,41)
41 FORMAT(lx,'MODEL TIME UNIT IS DAYS')
GO TO 100
50 WRITE(IOUT,51)
51 FORMAT(lx,'MODEL TIME UNIT IS YEARS')
C
C6------READ & PRINT INPUT UNIT NUMBERS (IUNIT) FOR MAJOR OPTIONS.
100 READ(INBAS,101) IUNIT
101 FORMAT(l24A13)
WRITE(IOUT,102) (1,1=1,24),IUNIT
102 FORMAT(lhI0,'I/O UNITS:'lhI0, 'ELEMENT OF IUNIT:',2413,
1       '/lx', 'I/O UNIT:',2413)
C
C7------INITIALIZE TOAL ELAPSED TIME COUNTER STORAGE ARRAY COUNTER
C7------AND CALCULATE NUMBER OF CELLS.
TOTIM=0.
ISUMI=1
NODES=NLAY*NROW*NLAY
C
C8------RETURN
RETURN
END
### List of Variables for Module BAS1DF

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>INBAS</td>
<td>Package</td>
<td>Primary unit number from which input to the BAS1 Package will be read. INBAS = 1.</td>
</tr>
<tr>
<td>IOUT</td>
<td>Global</td>
<td>Primary unit number for all printed output. IOUT = 6.</td>
</tr>
<tr>
<td>ISUM</td>
<td>Global</td>
<td>Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.</td>
</tr>
<tr>
<td>ITMUNI</td>
<td>Package</td>
<td>Code for time units for this problem:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - undefined</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - hours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 - days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - years</td>
</tr>
<tr>
<td>IUNIT</td>
<td>Module</td>
<td>DIMENSION (24), Primary input units for each of the major options.</td>
</tr>
<tr>
<td>HEADNG</td>
<td>Package</td>
<td>DIMENSION (32), Heading printed on output to identify the problem.</td>
</tr>
<tr>
<td>NCOL</td>
<td>Global</td>
<td>Number of columns in the grid.</td>
</tr>
<tr>
<td>NLAY</td>
<td>Global</td>
<td>Number of layers in the grid.</td>
</tr>
<tr>
<td>NODES</td>
<td>Global</td>
<td>Number of cells (nodes) in the finite-difference grid.</td>
</tr>
<tr>
<td>NPER</td>
<td>Global</td>
<td>Number of stress periods.</td>
</tr>
<tr>
<td>NROW</td>
<td>Global</td>
<td>Number of rows in the grid.</td>
</tr>
<tr>
<td>TOTIM</td>
<td>Package</td>
<td>Elapsed time in the simulation.</td>
</tr>
</tbody>
</table>
Module BASIAL allocates space for data arrays used by the BAS Package. Space is allocated for HNEW, HOLD, IBOUND, CR, CC, CV, HCOF, RHS, DELR, DELC, and IOFLG. Space is allocated for the STRT array if the user intends to calculate drawdown. Space is also allocated for an array called BUFFER, which is used to accumulate various data arrays such as drawdown and cell-by-cell flow terms when they are being calculated prior to output. To conserve space, the user may specify that arrays BUFFER and RHS should occupy the same space.

The number of spaces allocated for each of the arrays—HOLD, IBOUND, CR, CC, CV, HCOF, RHS, STRT, and BUFFER is equal to the number of cells in the grid. Twice that number of spaces is reserved for HNEW because it is double precision. DELR and DELC are allocated a number of spaces equal to the number of rows and columns, respectively. IOFLG (an array of flags used by Output Control) is allocated a number of spaces equal to four times the number of layers.

Module BASIAL performs its functions in the following order:

1. Print a message identifying the package.

2. Read and print flags IAPART and ISTRT which indicate whether the BUFFER and RHS arrays should occupy the same space and whether the start array (STRT) should be saved.

3. Store in ISOLD the location in the X array of the first unallocated space. Calculate the number of cells in the grid.

4. Allocate space for HNEW, HOLD, IBOUND, CR, CC, CV, HCOF, RHS, DELR, DELC, and IOFLG.

5. If the user specified that BUFFER and RHS should share space (IAPART equal to zero), set the address of the BUFFER (LCBUFF) equal to the address of RHS(LCRHS); otherwise, allocate separate space for BUFFER.

6. If the user specified that the starting array must be saved, allocate space for STRT.

7. Print the amount of space used by the BAS Package.

8. RETURN.
IAPART is a flag specified by the user which, if equal to zero, indicates that the arrays BUFFER and RHS should overlay each other.

BUFFER is an array in which data is temporarily stored while it is being gathered for printing.

RHS is an array which contains the right hand side of each finite-difference equation.

ISTRT is a flag specified by the user. If it is not equal to zero, starting heads are to be saved.

ISOLD marks the location of ISUM before any space was allocated by this module. After all space is allocated, ISOLD is subtracted from ISUM to calculate the amount of space allocated by this module.

ISUM is a counter which contains the location of the first unallocated element in the X array. Each time space is allocated for an array, the value in ISUM is incremented by the size of the array.
SUBROUTINE BASAL(ISUM, LEX, LOHNM, LOHOLD, LCIBOU, LCOOR, LCGC, LCOV,
1 LOHC, LORH, LODEL, LCSTR, LCOFF, LCIPL, IMBA, ISTR, NCOL, NROW, NLAY, IOUT)
C********** VERSION 1515 12/11/87 BASIC
C******************************************************************************
C ALLOCATE SPACE FOR BASIC MODEL ARRAYS
******************************************************************************
C
C SPECIFICATIONS:

C1------PRINT A MESSAGE IDENTIFYING THE PACKAGE.
WRITE(IOUT,1)IMBA
1 FORMAT(I10,'BASAL -- BASIC MODEL PACKAGE, VERSION 1, 12/11/87',
2' READ FROM UNIT ',I3)
C
C2------READ & PRINT FLAG IAPART (RHS & BUFFER SHARE SPACE?) AND
C2------FLAG ISTRT (SHOULD STARTING HEADS BE SAVED FOR DRAWDOWN)
READ(INBA,2) IAPART, ISTRT
2 FORMAT(2110)
IF(IAPART.EQ.0) WRITE(IOUT,3)
3 FORMAT(1X,'ARRAYS FOR S AND BUFFER WILL SHARE MEMORY. ')
IF(ISTRT.NE.0) WRITE(IOUT,4)
4 FORMAT(1X,'START HEAD WILL BE SAVED')
IF(ISTRT.EQ.0) WRITE(IOUT,5)
5 FORMAT(1X,'START HEA D WILL NOT BE SAVED',
1' DRAWDOWN CANNOT BE CALCULATED')
C
C3------STORE, IN ISOLD, LOCATION OF FIRST UNALLOCATED SPACE IN X.
ISOLD=ISUM
NROW=NCOL*NLAY
C
C4------ALLOCATE SPACE FOR ARRAYS.
LOHNM=ISUM
ISUM=ISUM+2*NCOL
LOHOLD=ISUM
ISUM=ISUM+NCOL
LCIBOU=ISUM
ISUM=ISUM+NCOL
LOHR=ISUM
ISUM=ISUM+NCOL
LCOOR=ISUM
ISUM=ISUM+NCOL
LCSTR=ISUM
ISUM=ISUM+NCOL
LCIPL=ISUM
ISUM=ISUM+NCOL
ISUM=ISUM+NROW
LCIOF=ISUM
ISUM=ISUM+4*NLAY
C
C5------IF BUFFER AND RHS SHARE SPACE THEN LCIPL+LCOER.
LCBOU=ISUM
LCBOU=LCBOU+1
LORH=ISUM
ISUM=ISUM+NCOL
C
C6------IF ISTRT WILL BE SAVED THEN ALLOCATE SPACE.
50 LCSTR=ISUM
IF(ISTRT.NE.0) ISUM=ISUM+NCOL
ISP=ISUM-ISOLD
C
C7------PRINT AMOUNT OF SPACE USED.
WRITE(IOUT,6) ISP
6 FORMAT(I10,1' ELEMENTS IN X ARRAY ARE USED BY BASAL'
1' ISUM=ISUM-1
WRITE(IOUT,7) ISUM, LEX
7 FORMAT(I10,1' ELEMENTS OF X ARRAY USED OUT OF ',I8)
1' ISUM=ISUM-LENX
WRITE(IOUT,8)
8 FORMAT(I10,' *** X ARRAY MUST BE DIMENSIONED LARGER'*)
C
C8------RETURN
RETURN
C
END
### List of Variables for Module BASIAL

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAPART</td>
<td>Module</td>
<td>Flag set by user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0, arrays RHS and BUFFER will share space in the X array.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠ 0, arrays RHS and BUFFER will not share space in the X array.</td>
</tr>
<tr>
<td>INBAS</td>
<td>Package</td>
<td>Primary unit number from which input to the BASI Package will be read. INBAS = 1.</td>
</tr>
<tr>
<td>IOUT</td>
<td>Global</td>
<td>Primary unit number for all printed output. IOUT = 6.</td>
</tr>
<tr>
<td>ISOLD</td>
<td>Package</td>
<td>Before this module allocates space, ISOLD is set equal to ISUM. After allocation, ISOLD is subtracted from ISUM to get ISP, the amount of space in the X array allocated by this module.</td>
</tr>
<tr>
<td>ISP</td>
<td>Module</td>
<td>Number of words in the X array allocated by this module.</td>
</tr>
<tr>
<td>ISTRRT</td>
<td>Package</td>
<td>Flag.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠ 0, starting heads will be saved so that drawdown can be calculated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0, starting heads will not be saved.</td>
</tr>
<tr>
<td>ISUM</td>
<td>Global</td>
<td>Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.</td>
</tr>
<tr>
<td>ISUM1</td>
<td>Module</td>
<td>Index number of the last element of the X array allocated by this module.</td>
</tr>
<tr>
<td>LCBUFF</td>
<td>Package</td>
<td>Location in the X array of the first element of array BUFF.</td>
</tr>
<tr>
<td>LCCC</td>
<td>Package</td>
<td>Location in the X array of the first element of array CC.</td>
</tr>
<tr>
<td>LCCR</td>
<td>Package</td>
<td>Location in the X array of the first element of array CR.</td>
</tr>
<tr>
<td>LCCV</td>
<td>Package</td>
<td>Location in the X array of the first element of array CV.</td>
</tr>
<tr>
<td>LCDEL</td>
<td>Package</td>
<td>Location in the X array of the first element of array DELC.</td>
</tr>
<tr>
<td>LCDELNR</td>
<td>Package</td>
<td>Location in the X array of the first element of array DELR.</td>
</tr>
<tr>
<td>LCHCOF</td>
<td>Package</td>
<td>Location in the X array of the first element of array HCOF.</td>
</tr>
<tr>
<td>LCHNEW</td>
<td>Package</td>
<td>Location in the X array of the first element of array HNEW.</td>
</tr>
<tr>
<td>LCHOLD</td>
<td>Package</td>
<td>Location in the X array of the first element of array HOLD.</td>
</tr>
<tr>
<td>LCIIOU</td>
<td>Package</td>
<td>Location in the X array of the first element of array IBOUND.</td>
</tr>
<tr>
<td>LCIOFL</td>
<td>Package</td>
<td>Location in the X array of the first element of array IOFLG.</td>
</tr>
<tr>
<td>LCRHS</td>
<td>Package</td>
<td>Location in the X array of the first element of array RHS.</td>
</tr>
<tr>
<td>LCSTRT</td>
<td>Package</td>
<td>Location in the X array of the first element of array STRT.</td>
</tr>
<tr>
<td>LENX</td>
<td>Global</td>
<td>Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.</td>
</tr>
<tr>
<td>NCOL</td>
<td>Global</td>
<td>Number of columns in the grid.</td>
</tr>
<tr>
<td>NLAY</td>
<td>Global</td>
<td>Number of layers in the grid.</td>
</tr>
<tr>
<td>NRCL</td>
<td>Module</td>
<td>Number of cells in the grid.</td>
</tr>
<tr>
<td>NROW</td>
<td>Global</td>
<td>Number of rows in the grid.</td>
</tr>
</tbody>
</table>
Narrative for Module BAS1RP

This module reads and prepares data for the BAS Package. It reads the boundary array (IBOUND) and the starting-head array (HNEW), sets the heads in no-flow cells to a user-supplied value (for printout convenience), initializes the starting-head array (STRT) and the volumetric-budget accumulators (VBVL), and sets up the Output Control System. The IBOUND codes are as follows.

<table>
<thead>
<tr>
<th>Code</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>negative</td>
<td>constant head</td>
</tr>
<tr>
<td>zero</td>
<td>inactive (no-flow)</td>
</tr>
<tr>
<td>positive</td>
<td>variable head</td>
</tr>
</tbody>
</table>

The user must specify a head value HNOFLO that he wants printed for no-flow (inactive) cells. That value is only used during printing and makes inactive cells stand out on the listing (e.g., 0.0 and 9999.99).

Recall that initial heads are needed for each time step; however, they must be read for only the first time step, at which time they are called the starting heads. For subsequent time steps, the ending heads of the preceding time step will be used as the initial heads of the current time step. The starting heads are read in single precision into the array HOLD and converted to double precision as they are moved into HNEW.

Module BAS1RP performs its functions in the following order:

1. Print the simulation title and calculate the number of cells in a layer.
2. Read the boundary array (IBOUND).
3. Read and print the head value to be printed for no-flow cells (HNOFLO).
4. Read the starting heads into array HOLD.
5. Copy the starting heads (and convert to double precision) from HOLD into HNEW.
6. If the starting heads must be saved, copy them from HOLD to STRT.
7. Initialize volumetric-budget accumulators.
8. Call submodule SBASII to initialize the Output Control System.
9. RETURN.
Flow Chart for Module BAS1RP

HNOFLO is a value assigned to head in inactive (no-flow) cells. It makes those cells stand out in listings of heads.

HNEW is an array containing the latest estimates of heads. It starts each time step with heads calculated for the end of the previous time step. It is changed at each iteration until the last iteration when it contains the heads at the end of the time step.

HOLD is an array containing heads at the beginning of the current time step. At the beginning of a time step, HOLD AND HNEW contain identical values. HNEW changes from one iteration to the next; HOLD does not.

OUTPUT CONTROL is part of the Basic Package which gives the user the ability to control the kind and amount of information that is printed by the program.

U2DINT is a utility module which reads two-dimensional integer arrays.
SUBROUTINE BAS1RP(IBOUND,HNEW,STRT,HOLD,ISTRT,INBAS,
1 HEADING,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,
2 IDNFM,IHEDUN,IDNUN,IOUT)
C------VERSION 1628 15MAY1987 BAS1RP
C ******************************************************************************
C READ AND INITIALIZE BASIC MODEL ARRAYS
C ******************************************************************************
C SPECIFICATIONS:
C CHARACTER*4 HEADING,ANAME
C DOUBLE PRECISION HNEW,HNOFLO
C DIMENSION HNEW(NODES),IBOUND(NODES),STRT(NODES),HOLD(NODES),
1 ANAME(6,2),ANAME(6,2),ANAME(6,2),ANAME(6,2),ANAME(6,2),
WRITE(IOUT,1) HEADNG
1 FORMAT(lH1,32A4)  NCR=NCOL*NROW
C
C2------READ BOUNDARY ARRAY(IBOUND) ONE LAYER AT A TIME.
DO 100 K=1,NLAY
K=K  LOC=1+(K-1)*NCR
CALL U2DINT(IBOUND(LOC),ANAME(1,1),NROW,NCOL,KK,INBAS,IOUT)
100 CONTINUE
C
C3------READ AND PRINT HEAD VALUE TO BE PRINTED FOR NO-FLOW CELLS.
READ(INBAS,2) TMP
2 FORMAT(lO.0) HNOFLO=TMP
WRITE(IOUT,3) TMP 3 FORMAT(lH0,'AQUIFER HEAD WILL BE SET TO ',1PG11.5,
1 ' AT ALL NO-FLOW NODES (IBOUND=0).')
C
C4------READ STARTING HEADS.
DO 300 K=1,NLAY
K=K  LOC=1+(K-1)*NCR
CALL U2DREL(HOLD(LOC),ANAME(1,2),NROW,NCOL,KK,INBAS,IOUT)
300 CONTINUE
C
C5------COPY INITIAL HEADS FROM HOLD TO HNEW.
DO 400 I=1,NODES
HNEW(I)=HOLD(I)
IF(IBOUND(I).EQ.O) HNEW(I)=HNOFLO
400 CONTINUE
C
C6------IF STARTING HEADS ARE TO BE SAVED THEN COPY HOLD TO STRT.
IF(ISTRT.EQ.O) GO TO 590
DO 500 I=1,NODES
STRT(I)=HOLD(I)
500 CONTINUE
C
C7------INITIALIZE VOLUMETRIC BUDGET ACCUMULATORS TO ZERO.
590 DO 600 I=1,20
DO 600 J=1,4
VBVL(J,1)=O.
600 CONTINUE
C
C8------SET UP OUTPUT CONTROL.
CALL SBASIL(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM,
1 IDNFM,IHEDUN,IDNUN)
C
C9------RETURN
1000 RETURN
END
### List of Variables for Module BAS1RP

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANAME</td>
<td>Module</td>
<td>Label for printout of input array.</td>
</tr>
<tr>
<td>HEADNG</td>
<td>Package</td>
<td>DIMENSION (32), Heading printed on output to identify problem.</td>
</tr>
<tr>
<td>HNEW</td>
<td>Global</td>
<td>DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.</td>
</tr>
<tr>
<td>HNOFLO</td>
<td>Module</td>
<td>User specified value for head in cells which are inactive at the start of simulation.</td>
</tr>
<tr>
<td>HOLD</td>
<td>Global</td>
<td>DIMENSION (NCOL,NROW,NLAY), Head at the start of the current time step.</td>
</tr>
<tr>
<td>I</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>IBOUND</td>
<td>Global</td>
<td>DIMENSION (NCOL,NROW,NLAY), Status of each cell. &lt; 0, constant-head cell &lt; 0, variable-head cell = 0, inactive cell &gt; 0, variable-head cell</td>
</tr>
<tr>
<td>IDDNFM</td>
<td>Package</td>
<td>Code for format in which drawdown should be printed.</td>
</tr>
<tr>
<td>IDDNUN</td>
<td>Package</td>
<td>Unit number on which an unformatted record containing drawdown should be recorded.</td>
</tr>
<tr>
<td>IHEDFM</td>
<td>Package</td>
<td>Code for format in which head should be printed.</td>
</tr>
<tr>
<td>IHEDUN</td>
<td>Package</td>
<td>Unit number on which an unformatted record containing head should be recorded.</td>
</tr>
<tr>
<td>INBAS</td>
<td>Package</td>
<td>Primary unit number from which input to BAS1 Package will be read. INBAS = 1.</td>
</tr>
<tr>
<td>INOC</td>
<td>Package</td>
<td>Unit number from which input to output control option will be read. IOUT = 6.</td>
</tr>
<tr>
<td>IOFLG</td>
<td>Package</td>
<td>DIMENSION (NLAY,4), Flags to control printing and recording of head and drawdown for each layer. (NLAY,1) ≠ 0, heads will be printed.</td>
</tr>
<tr>
<td>IOUT</td>
<td>Global</td>
<td>Primary unit number for all printed output. IOUT = 6.</td>
</tr>
<tr>
<td>ISTRT</td>
<td>Package</td>
<td>Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.</td>
</tr>
<tr>
<td>J</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>K</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>KK</td>
<td>Module</td>
<td>Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument,</td>
</tr>
<tr>
<td>LOC</td>
<td>Module</td>
<td>Pointer to location in an array for a specific layer.</td>
</tr>
<tr>
<td>NCOL</td>
<td>Global</td>
<td>Number of columns in the grid.</td>
</tr>
<tr>
<td>NCR</td>
<td>Module</td>
<td>Number of cells in a layer.</td>
</tr>
<tr>
<td>NLAY</td>
<td>Global</td>
<td>Number of layers in the grid.</td>
</tr>
<tr>
<td>NODES</td>
<td>Global</td>
<td>Number of cells (nodes) in the finite-difference grid.</td>
</tr>
<tr>
<td>NROW</td>
<td>Global</td>
<td>Number of rows in the grid.</td>
</tr>
<tr>
<td>STRT</td>
<td>Package</td>
<td>DIMENSION (NCOL,NROW,NLAY), Starting head.</td>
</tr>
<tr>
<td>TMP</td>
<td>Module</td>
<td>Single-precision temporary storage place for HNOFLO.</td>
</tr>
<tr>
<td>VBVL</td>
<td>Global</td>
<td>DIMENSION (4,20), Entries for the volumetric budget.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For flow component N, the values in VBVL are: (1,N) Rate for current time step into the flow field. (2,N) Rate for current time step out of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the flow field. (3,N) Volume into the flow field during simulation. (4,N) Volume out of the flow field during simulation.</td>
</tr>
</tbody>
</table>

4-30
Module BAS1ST reads timing information for a stress period and initializes variables used to calculate the length of time steps and elapsed time. Each stress period is divided into time steps which form a geometric progression (for a stress period, there is a multiplier TSMULT such that the length of a time step is equal to TSMULT times the length of the previous time step). If the length of the stress period (PERLEN) and the number of time steps (NSTP) is known, the length of the first time step DELT can be calculated with the equation

\[ \text{DELT} = \frac{(1-\text{TSMULT}) \times \text{PERLEN}}{1-(\text{TSMULT})^\text{NSTP}}. \]

Note: When TSMULT is equal to one, all the time steps are the same length. In that case, the time-step length is the length of the stress period (PERLEN) divided by the number of time steps (NSTP).

Module BAS1ST performs its functions in the following order:

1. Read the length of the stress period (PERLEN), the number of time steps in the stress period (NSTP), and the time-step multiplier (TSMULT).
2. Calculate the length of the first time step.
   (a) Assume the time-step multiplier is equal to one.
   (b) If the time-step multiplier (TSMULT) is not equal to one, calculate the first term of the geometric progression.
3. Print the timing information.
4. Initialize the variable PERTIM which keeps track of elapsed time within a stress period.
5. RETURN.
PERLEN is the length of a stress period.

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

TSMULT is a constant which, when multiplied by the length of a time step, gives the length of the next time step.

DELT is the length of the first time step. Since the time steps form a geometric progression, the formula for calculating DELT is:

$$DELT = \frac{(1-TSMULT) \times PERLEN}{(1-TSMULT^{NSTP})}$$

PERTIM is a field in which elapsed time during a stress period is accumulated. During each time step, the length of the time step is added to PERTIM.
SUBROUTINE BASIST(NSTP, DELT, TSMULT, PERTIM, KPER, INBAS, IOUT)

C
C------VERSION 1614 08SEP1982 BASIST
C *************************************************************************************************
C SETUP TIME PARAMETERS FOR NEW TIME PERIOD
C *************************************************************************************************
C
C SPECIFICATIONS:
-----------------------------------------------
-----------------------------------------------
-----------------------------------------------
C1-------READ LENGTH OF STRESS PERIOD, NUMBER OF TIME STEPS AND.
C1-------TIME STEP MULTIPLIER.
   READ (INBAS,1) PERLEN,NSTP, TSMULT
     1 FORMAT(F10.0,I10,F10.0)
C
C2------CALCULATE THE LENGTH OF THE FIRST TIME STEP.
C  
C2A-----ASSUME TIME STEP MULTIPLIER IS EQUAL TO ONE.
   DELT=PERLEN/FLOAT(NSTP)
C
C2B-----IF TIME STEP MULTIPLIER IS NOT ONE THEN CALCULATE FIRST
C2B-----TERM OF GEOMETRIC PROGRESSION.
   IF(TSMULT.NE.1.) DELT=PERLEN*(1.-TSMULT)/(1.-TSMULT**NSTP)
C
C3------PRINT TIMING INFORMATION.
   WRITE (IOUT,2) KPER,PERLEN,NSTP, TSMULT, DELT
     2 FORMAT(1HI,51X,'STRESS PERIOD NO.',I4,' LENGTH =',G15.7/52X
       1,46('-')//52X,'NUMBER OF TIME STEPS =',I6
       2//53X,'MULTIPLIER FOR DELT =',F10.3
       3//50X,'INITIAL TIME STEP SIZE =',G15.7)
C
C4------INITIALIZE PERTIM (ELAPSED TIME WITHIN STRESS PERIOD).
   PERTIM=0.
C
C5------RETURN
   RETURN
END
### List of Variables for Module BASIST

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELT</td>
<td>Global</td>
<td>Length of the current time step.</td>
</tr>
<tr>
<td>INBAS</td>
<td>Package</td>
<td>Primary unit number from which input to the BAS1 Package will be read. INBAS = 1.</td>
</tr>
<tr>
<td>IOUT</td>
<td>Global</td>
<td>Primary unit number for all printed output. IOUT = 6.</td>
</tr>
<tr>
<td>KPER</td>
<td>Global</td>
<td>Stress period counter.</td>
</tr>
<tr>
<td>NSTP</td>
<td>Global</td>
<td>Number of time steps in the current stress period.</td>
</tr>
<tr>
<td>PERLEN</td>
<td>Module</td>
<td>Length of the stress period.</td>
</tr>
<tr>
<td>PERTIM</td>
<td>Package</td>
<td>Elapsed time during the current stress period.</td>
</tr>
<tr>
<td>TSMULT</td>
<td>Package</td>
<td>Multiplier to get from one time step length to the next.</td>
</tr>
</tbody>
</table>
Narrative for Module BAS1AD

Module BAS1AD calculates the length of the time step, accumulates the elapsed time for the stress period and the total simulation period, and sets the old head values equal to the new head values.

Within a stress period, the length of the time steps form a geometric progression—the length of each time step is a constant (TSMULT) times the length of the previous time step. The length of the first time step is calculated in module BAS1ST.

The array HNEW contains the heads calculated for the end of the last time step. Those heads which are also the heads at the beginning of the current time step are copied into HOLD.

Module BAS1AD performs its functions in the following order:

1. If this is not the first time step in the stress period, calculate the length of the time step (DELT). Note: The length of the first time step is calculated by BAS1ST.

2. Accumulate the elapsed time since the beginning of the simulation period (TOTIM) and the beginning of the stress period (PERTIM).

3. Set the heads at the beginning of this time step (HOLD) equal to the heads at the end of the previous time step (HNEW).

4. RETURN.
TOTIM is an accumulator in which the total elapsed time since the beginning of the simulation is stored.

PERTIM is an accumulator in which the total elapsed time during the current stress period is stored.

HOLD is the head distribution at the beginning of a time step.

HNEW is the head distribution at the end of a time step.
SUBROUTINE BAS1AD(DELT, TSMULT, TOTIM, PERTIM, HNEW, HOLD, KSTP, NCOL, NROW, NLAY)

C C-----VERSION 1412 22FEB1982 BAS1AD
C C******************************************************************************
C C ADVANCE TO NEXT TIME STEP
C******************************************************************************
C C SPECIFICATIONS:
C******************************************************************************
C DOUBLE PRECISION HNEW
C DIMENSION HNEW(NCOL, NROW, NLAY), HOLD(NCOL, NROW, NLAY)
C******************************************************************************
C IF NOT FIRST TIME STEP THEN CALCULATE TIME STEP LENGTH.
C IF(KSTP.NE.1) DELT=TSMULT*DELT
C
C ACCUMULATE ELAPSED TIME IN SIMULATION(TOTIM) AND IN THIS
C STRESS PERIOD(PERTIM).
C TOTIM=TOTIM+DELT
C PERTIM=PERTIM+DELT
C
C COPY HNEW TO HOLD.
C DO 10 K=1,NLAY
C DO 10 I=1,NROW
C DO 10 J=1,NCOL
C 10 HOLD(J,I,K)=HNEW(J,I,K)
C
C RETURN
C RETURN
CEND
<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELT</td>
<td>Global</td>
<td>Length of the current time step.</td>
</tr>
<tr>
<td>HNEW</td>
<td>Global</td>
<td>DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.</td>
</tr>
<tr>
<td>HOLD</td>
<td>Global</td>
<td>DIMENSION (NCOL,NROW,NLAY), Head at the start of the current time step.</td>
</tr>
<tr>
<td>I</td>
<td>Module</td>
<td>Row index.</td>
</tr>
<tr>
<td>J</td>
<td>Module</td>
<td>Column index.</td>
</tr>
<tr>
<td>K</td>
<td>Module</td>
<td>Layer index.</td>
</tr>
<tr>
<td>KSTP</td>
<td>Global</td>
<td>Time step counter. Reset at the start of each stress period.</td>
</tr>
<tr>
<td>NCOL</td>
<td>Global</td>
<td>Number of columns in the grid.</td>
</tr>
<tr>
<td>NLAY</td>
<td>Global</td>
<td>Number of layers in the grid.</td>
</tr>
<tr>
<td>NROW</td>
<td>Global</td>
<td>Number of rows in the grid.</td>
</tr>
<tr>
<td>PERTIM</td>
<td>Package</td>
<td>Elapsed time during the current stress period.</td>
</tr>
<tr>
<td>TOTIM</td>
<td>Package</td>
<td>Elapsed time in the simulation.</td>
</tr>
<tr>
<td>TSMULT</td>
<td>Package</td>
<td>Multiplier to get from one time step length to the next.</td>
</tr>
</tbody>
</table>
Narrative for Module BAS1FM

This module initializes the arrays in which the right hand side (RHS) and the h-coefficient (HCOF) are accumulated.

Recall that the equation for cell i,j,k contains a term RHS_{i,j,k} on the right hand side and a coefficient HCOF_{i,j,k} (h-coefficient) which multiplies h_{i,j,k} on the left hand side of the equation. The right-hand-side term and the h-coefficient are the sum of terms related to many of the flow components. They are calculated every time the equations are formulated.

Module BAS1FM performs its functions in the following order:

1. For each cell, initialize (set equal to zero) the HCOF and RHS accumulators.

2. RETURN.
SUBROUTINE BASLFM(HCOF, RHS, NODES)

C

C-------VERSION 1632 24JUL1987 BASLFM
C
C   ***********************************************************************
C   SET HCOF=RHS=0.
C   ***********************************************************************
C
C SPECIFICATIONS:
C
C------------------------------------------------------------------------
DIMENSION HCOF(NODES), RHS(NODES)
C------------------------------------------------------------------------
C
C1-------FOR EACH CELL INITIALIZE HCOF AND RHS ACCUMULATORS.
      DO 100 I=1,NODES
         HCOF(I)=0.
         RHS(I)=0.
      100 CONTINUE
C
C2-------RETURN
      RETURN
      END
List of Variables for Module BAS1FM

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>HCOF</td>
<td>Global</td>
<td>DIMENSION (NODES). Coefficient of head in cell (J,I,K) in the finite-difference equation.</td>
</tr>
<tr>
<td>NODES</td>
<td>Global</td>
<td>Number of cells (nodes) in the finite-difference grid.</td>
</tr>
<tr>
<td>RHS</td>
<td>Global</td>
<td>DIMENSION (NODES), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.</td>
</tr>
</tbody>
</table>
Narrative for Module BAS1OC

Module BAS1OC sets flags used by the budget and output procedures to determine what data should be printed or recorded on disk. There are three individual flags and one table of flags. The individual flags are IHDDFL which indicates that head or drawdown is to be printed or recorded, IBUDFL which indicates that the overall budget should be printed, and ICBCFL which indicates that cell-by-cell flow terms should be calculated and printed or recorded. The table of flags called IOFLG has four flags for each layer. They correspond to the four options: print heads, print drawdown, save heads, and save drawdown. The flags in IOFLG are used in conjunction with the flag IHDDFL. If IHDDFL is set, IOFLG is used to determine head and drawdown on a layer-by-layer basis. If IHDDFL is not set, heads and drawdown are not printed or saved and IOFLG is ignored.

If the user is controlling output, the flags are read at each time step; if not, IOFLG is set at the start of the simulation and the individual flags are set at each time step.

Module BAS1OC performs its functions in the following order:

1. Determine if the user has specified that he will control output. He does so by coding a positive integer in the twelfth element of the IUNIT table. That integer is read by module BAS1DF and is passed to this module (BAS1OC) under the name INOC. Go to either 2 or 3.

2. The user is not controlling output. Set flags for default-output and then return. Flags IHDDFL and IBUDFL are set only at the last time step in each stress period or when the iterative procedure fails to converge. RETURN.

3. The user has chosen to control output. Read and print the code INCODE and flags IHDDFL, IBUDFL, and ICBCFL. The code INCODE gives the user several options for specifying the flag table IOFLG.

4. Determine whether INCODE is less than zero, equal to zero, or greater than zero. Go to 5, 6, or 7.

5. INCODE is less than zero. Use the IOFLG flags used in the previous time step and print a message to that effect. Go to 8.

6. INCODE is equal to zero. Read IOFLG for layer 1 and then set flags in all other layers equal to those in layer 1. Go to 8.

7. INCODE is greater than zero. Read IOFLG array. Go to 8.

8. Regardless of what the user has specified, set the flag IBUDFL if the iterative procedure failed to converge or if the current time step is the last time step in the stress period.

9. RETURN.
INOC is the input unit for Output Control specifications. It is specified by the user as the twelfth element of the IUNIT array. If it is less than or equal to zero, the user has chosen the default output. If it is greater than zero, the user has chosen to control output.

INCODE provides the user with options for filling the IOFLG array.

If INCODE < 0, IOFLG from the last time step is reused.

If INCODE = 0, IOFLG for layer 1 is read and all other layers are set equal to layer 1.

If INCODE > 0, IOFLG is read.

IOFLG is a table of flags with one entry for each layer. Each entry has four flags:

1 --- head print
2 --- drawdown print
3 --- head save
4 --- drawdown save

If a flag is set (equal to 1), head or drawdown for the corresponding layer is either printed or saved on disk.

IHDDFL is the head/drawdown flag. If it is set, heads and drawdowns will be written in accordance with the flags in IOFLG.

IBUDFL is the budget print flag. If it is set, the overall budget will be printed.

ICBCFL is the cell-by-cell flow term flag. If it is set, cell-by-cell flow terms will be printed or recorded on disk for those components of flow for which the CBC flag (IWELCB, IRCHCB, IDRNCB, etc.) is set.
SUBROUTINE BASLCC(NSTP,KSTP,ICNVG,IOFLG,NLAY,
IBUDFL,ICBCFL,IHDDFL,INOC,IOUT)
C
C--------VERSION 1632 24JUL1987 BASLCC
C ******t**************************************~****************~***
C  *********************************************~***~****************
C
C SPECIFICATIONS:
C
C DIMENSION IOFLG(NLAY,4)
C
C1------TEST UNIT NUMBER (INOC (INOC=UNIT(12)) TO SEE IF
C1------OUTPUT CONTROL IS ACTIVE.
C IF(INOC.NE.0)GO TO 500
C
C2------IF OUTPUT CONTROL IS INACTIVE THEN SET DEFAULTS AND RETURN.
C IHDDFL=U
C IF(INCVG.EQ.0 .OR. KSTP.EQ.NSTP)IHDDFL=1
C IBUDFL=0
C IF(INCVG.EQ.0 .OR. KSTP.EQ.NSTP)IBUDFL=1
C ICBCFL=0
C GO TO 1000
C
C3-------READ AND PRINT OUTPUT FLAGS AND CODE FOR DEFINING IOFLG.
C 500 READ(INOC,1) INCODE,IHDDFL,IBUDFL,ICBCFL
C 1 FORMAT(4110)
C WRITE(IOUT,3) IHDDFL,IBUDFL,ICBCFL
C 3 FORMAT(1HO,'HEAD/DRAWDOWN PRINTOUT FLAG =',I2,
C 1X,'TOTAL BUDGET PRINTOUT FLAG =',I2,
C 2SX,'CELL-BY-CELL FLOW TERM FLAG =',I2)
C
C4------DECODE INCODE TO DETERMINE HOW TO SET FLAGS IN IOFLG.
C IF(INCODE) 100,200,300
C
C5------USE IOFLG FROM LAST TIME STEP.
C 100 WRITE(IOUT,101)
C 101 FORMAT(1H,'REUSING PREVIOUS VALUES OF IOFLG')
C GO TO 600
C
C6------READ IOFLG FOR LAYER 1 AND ASSIGN SAME TO ALL LAYERS
C 200 READ(INOC,201) (IOFLG(K,I),I=1,4),K=1,NLAY)
C 201 FORMAT(4110)
C DO 210 K=1,NLAY
C IOFLG(K,1)=IOFLG(1,1)
C IOFLG(K,2)=IOFLG(1,2)
C IOFLG(K,3)=IOFLG(1,3)
C IOFLG(K,4)=IOFLG(1,4)
C 210 CONTINUE
C WRITE(IOUT,211) (IOFLG(K,I),I=1,4),K=1,NLAY)
C 211 FORMAT(1H,'OUTPUT FLAGS FOR EACH LAYER: '/
C 1 1X,'HEAD DRAWDOWN HEAD DRAWDOWN'/
C 2 1X,'PRINTOUT PRINTOUT SAVE SAVE'/
C 3 1X,3A(-1)/1X,15,I10,18,18)
C GO TO 600
C
C7------READ IOFLG IN ENTIRETY
C 300 READ(INOC,301) ((IOFLG(K,I),I=1,4),K=1,NLAY)
C 301 FORMAT(4110)
C WRITE(IOUT,302)
C 302 FORMAT(1H,'OUTPUT FLAGS FOR EACH LAYER: '/
C 1 1X,'HEAD DRAWDOWN HEAD DRAWDOWN'/
C 2 1X,'LAYER PRINTOUT PRINTOUT SAVE SAVE'/
C 3 1X,3A(-1))
C WRITE(IOUT,303) (K,(IOFLG(K,I),I=1,4),K=1,NLAY)
C 303 FORMAT(1X,14,I10,110,18,18)
C
C8------THE LAST STEP IN A STRESS PERIOD AND STEPS WHERE ITERATIVE
C PROCEDURE FAILED TO CONVERGE GET A VOLUMETRIC BUDGET.
C 600 IF(INCVG.EQ.0 .OR. KSTP.EQ.NSTP) IBUDFL=1
C
C9------RETURN
C 1000 RETURN
END
List of Variables for Module BAS10C

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>IBUDFL</td>
<td>Package</td>
<td>Flag.</td>
</tr>
<tr>
<td>ICBCFL</td>
<td>Global</td>
<td>Flag.</td>
</tr>
<tr>
<td>ICNVG</td>
<td>Global</td>
<td>Flag is set equal to 1 when the iteration procedure has converged.</td>
</tr>
<tr>
<td>IHDDFL</td>
<td>Package</td>
<td>Flag.</td>
</tr>
<tr>
<td>INCODE</td>
<td>Module</td>
<td>Code specified by user.</td>
</tr>
<tr>
<td>INOC</td>
<td>Package</td>
<td>Unit number from which input to output control option will be read.</td>
</tr>
<tr>
<td>IOFLG</td>
<td>Package</td>
<td>DIMENSION (NLAY,4), Flags to control printing and recording of head and drawdown for each layer.</td>
</tr>
<tr>
<td>IOUT</td>
<td>Global</td>
<td>Primary unit number for all printed output. IOUT = 6.</td>
</tr>
<tr>
<td>K</td>
<td>Module</td>
<td>Layer index.</td>
</tr>
<tr>
<td>KSTP</td>
<td>Global</td>
<td>Time step counter. Reset at the start of each stress period.</td>
</tr>
<tr>
<td>M</td>
<td>Module</td>
<td>Index.</td>
</tr>
<tr>
<td>NLAY</td>
<td>Global</td>
<td>Number of layers in the grid.</td>
</tr>
<tr>
<td>NSTP</td>
<td>Global</td>
<td>Number of time steps in the current stress period.</td>
</tr>
</tbody>
</table>