USGS
science for a changing world

U.S. DEPARTMENT OF THE INTERIOR
March 3, 1849

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

Click here to return to USGS Publications

Module Documentation for the Basic Package

The Basic Package (BAS1) 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 intializes 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.

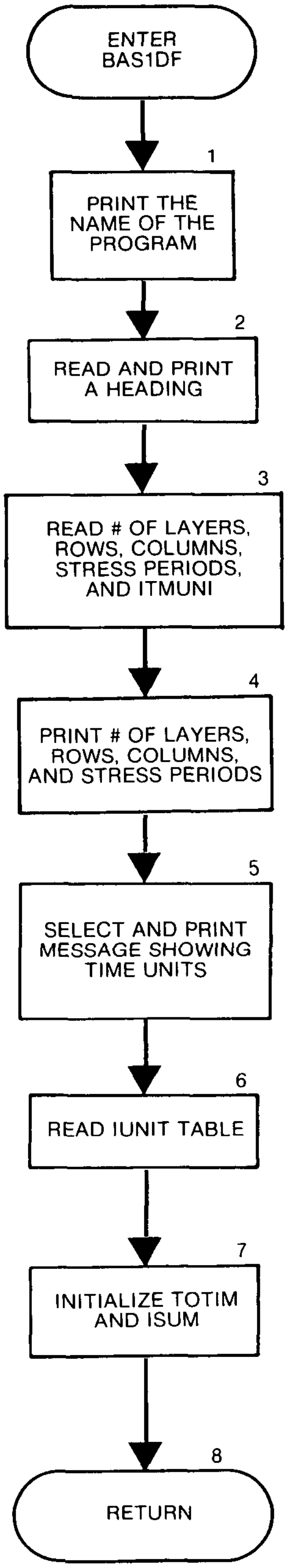
Flow Chart for Module BAS1DF

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.


ENTER BAS1DF
1
PRINT THE NAME OF THE PROGRAM
2
READ AND PRINT A HEADING
3
READ # OF LAYERS, ROWS, COLUMNS, STRESS PERIODS, AND ITMUNI
4
PRINT # OF LAYERS, ROWS, COLUMNS, AND STRESS PERIODS
5
SELECT AND PRINT MESSAGE SHOWING TIME UNITS
6
READ IUNIT TABLE
7
INITIALIZE TOTIM AND ISUM
8
RETURN

```
      SUBROUTINE BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,
     1       NLAY,NODES,INBAS,IOUT,IUNIT)
C
C-----VERSION 1513 12MAY1987 BAS1DF
C     ******************************************************************
C     DEFINE KEY MODEL PARAMETERS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 HEADNG
      DIMENSION HEADNG(32),IUNIT(24)
C     ------------------------------------------------------------------
C
C1------PRINT THE NAME OF THE PROGRAM.
      WRITE(IOUT,1)
    1 FORMAT(1H1,20X,'U.S. GEOLOGICAL SURVEY MODULAR',
     1      ' FINITE-DIFFERENCE GROUND-WATER MODEL')
C
C2------READ AND PRINT A HEADING.
      READ(INBAS,2) HEADNG
    2 FORMAT(20A4)
      WRITE(IOUT,3) HEADNG
    3 FORMAT(1H0,32A4)
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(8I10)
C
C4------PRINT # OF LAYERS, ROWS, COLUMNS AND STRESS PERIODS.
      WRITE(IOUT,5) NLAY,NROW,NCOL
    5 FORMAT(1X,I4,' LAYERS',I10,' ROWS',I10,' COLUMNS')
      WRITE(IOUT,6) NPER
    6 FORMAT(1X,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(1X,'MODEL TIME UNITS ARE UNDEFINED')
      GO TO 100
   10 WRITE(IOUT,11)
   11 FORMAT(1X,'MODEL TIME UNIT IS SECONDS')
      GO TO 100
   20 WRITE(IOUT,21)
   21 FORMAT(1X,'MODEL TIME UNIT IS MINUTES')
      GO TO 100
   30 WRITE(IOUT,31)
   31 FORMAT(1X,'MODEL TIME UNIT IS HOURS')
      GO TO 100
   40 WRITE(IOUT,41)
   41 FORMAT(1X,'MODEL TIME UNIT IS DAYS')
      GO TO 100
   50 WRITE(IOUT,51)
   51 FORMAT(1X,'MODEL TIME UNIT IS YEARS')
C
C6------READ & PRINT INPUT UNIT NUMBERS (IUNIT) FOR MAJOR OPTIONS.
  100 READ(INBAS,101) IUNIT
  101 FORMAT(24I3)
      WRITE(IOUT,102) (I,I=1,24),IUNIT
  102 FORMAT(1H0,'I/O UNITS:'/1X,'ELEMENT OF IUNIT:',24I3,
     1                      /1X,'        I/O UNIT:',24I3)
C
C7------INITIALIZE TOAL ELAPSED TIME COUNTER STORAGE ARRAY COUNTER
C7------AND CALCULATE NUMBER OF CELLS.
      TOTIM=0.
      ISUM=1
      NODES=NCOL*NROW*NLAY
C
C8------RETURN
      RETURN
      END
```

List of Variables for Module BAS1DF

Variable	Range	Definition
I	Module	Index.
INBAS	Package	Primary unit number from which input to the BAS1 Package will be read. INBAS = 1.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISUM	Global	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.
ITMUNI	Package	Code for time units for this problem: 0 - undefined 1 - seconds 2 - minutes 3 - hours 4 - days 5 - years
IUNIT	Module	DIMENSION (24), Primary input units for each of the major options.
HEADNG	Package	DIMENSION (32), Heading printed on output to identify the problem.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NODES	Global	Number of cells (nodes) in the finite-difference grid.
NPER	Global	Number of stress periods.
NROW	Global	Number of rows in the grid.
TOTIM	Package	Elapsed time in the simulation.

Narrative for Module BAS1AL

Module BAS1AL 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 BAS1AL 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.

Flow Chart for Module BAS1AL

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

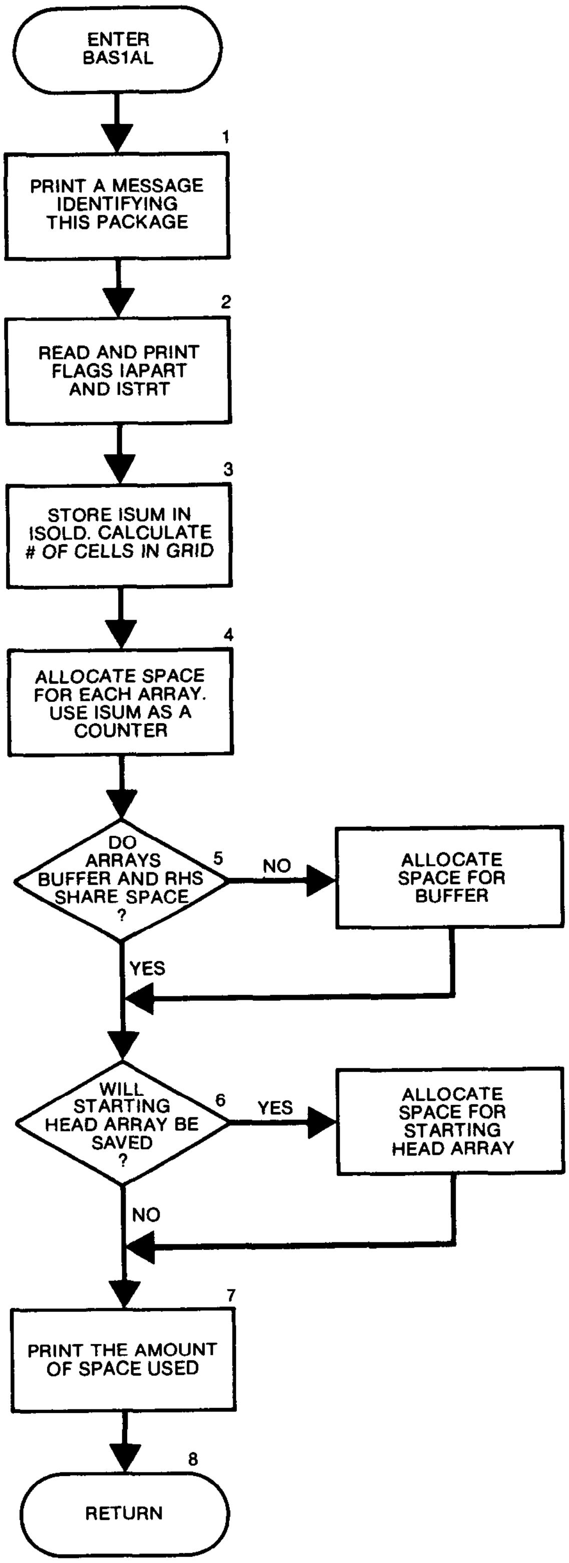
BUFFER is an array in which data is temporarily stored while it is being gathered for printing.

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.


ENTER BAS1AL
1
PRINT A MESSAGE IDENTIFYING THIS PACKAGE
2
READ AND PRINT FLAGS IAPART AND ISTRT
3
STORE ISUM IN ISOLD. CALCULATE # OF CELLS IN GRID
4
ALLOCATE SPACE FOR EACH ARRAY. USE ISUM AS A COUNTER
5
DO ARRAYS BUFFER AND RHS SHARE SPACE ?
NO
ALLOCATE SPACE FOR BUFFER
YES
6
WILL STARTING HEAD ARRAY BE SAVED ?
YES
ALLOCATE SPACE FOR STARTING HEAD ARRAY
NO
7
PRINT THE AMOUNT OF SPACE USED
8
RETURN

```
      SUBROUTINE BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1            LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,INBAS,
     1            ISTRT,NCOL,NROW,NLAY,IOUT)
C-----VERSION 1515 12MAY1987 BAS1AL
C     ******************************************************************
C     ALLOCATE SPACE FOR BASIC MODEL ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------PRINT A MESSAGE IDENTIFYING THE PACKAGE.
      WRITE(IOUT,1)INBAS
    1 FORMAT(1H0,'BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 9/1/87',
     2' INPUT 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(INBAS,2) IAPART,ISTRT
    2 FORMAT(2I10)
      IF(IAPART.EQ.0) WRITE(IOUT,3)
    3 FORMAT(1X,'ARRAYS RHS AND BUFF 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 HEAD WILL NOT BE SAVED',
     1      ' -- DRAWDOWN CANNOT BE CALCULATED')
C
C3------STORE,IN ISOLD, LOCATION OF FIRST UNALLOCATED SPACE IN X.
      ISOLD=ISUM
      NRCL=NROW*NCOL*NLAY
C
C4------ALLOCATE SPACE FOR ARRAYS.
      LCHNEW=ISUM
      ISUM=ISUM+2*NRCL
      LCHOLD=ISUM
      ISUM=ISUM+NRCL
      LCIBOU=ISUM
      ISUM=ISUM+NRCL
      LCCR=ISUM
      ISUM=ISUM+NRCL
      LCCC=ISUM
      ISUM=ISUM+NRCL
      LCCV=ISUM
      ISUM=ISUM+NROW*NCOL*(NLAY-1)
      LCHCOF=ISUM
      ISUM=ISUM+NRCL
      LCRHS=ISUM
      ISUM=ISUM+NRCL
      LCDELR=ISUM
      ISUM=ISUM+NCOL
      LCDELC=ISUM
      ISUM=ISUM+NROW
      LCIOFL=ISUM
      ISUM=ISUM+NLAY*4
C
C5------IF BUFFER AND RHS SHARE SPACE THEN LCBUFF=LCRHS.
      LCBUFF=LCRHS
      IF(IAPART.EQ.0) GO TO 50
      LCBUFF=ISUM
      ISUM=ISUM+NRCL
C
C6------IF STRT WILL BE SAVED THEN ALLOCATE SPACE.
   50 LCSTRT=ISUM
      IF(ISTRT.NE.0) ISUM=ISUM+NRCL
      ISP=ISUM-ISOLD
C
C7------PRINT AMOUNT OF SPACE USED.
      WRITE(IOUT,6) ISP
    6 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED BY BAS')
      ISUM1=ISUM-1
      WRITE(IOUT,7) ISUM1,LENX
    7 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,8)
    8 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C
C8------RETURN
      RETURN
C
      END
```

List of Variables for Module BAS1AL

Variable	Range	Definition
IAPART	Module	Flag set by user. = 0, arrays RHS and BUFFER will share space in the X array. ≠ 0, arrays RHS and BUFFER will not share space in the X array.
INBAS	Package	Primary unit number from which input to the BAS1 Package will be read. INBAS = 1.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISOLD	Package	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.
ISP	Module	Number of words in the X array allocated by this module.
ISTRT	Package	Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
ISUM	Global	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.
ISUM1	Module	Index number of the last element of the X array allocated by this module.
LCBUFF	Package	Location in the X array of the first element of array BUFF.
LCCC	Package	Location in the X array of the first element of array CC.
LCCR	Package	Location in the X array of the first element of array CR.
LCCV	Package	Location in the X array of the first element of array CV.
LCDELC	Package	Location in the X array of the first element of array DELC.
LCDELR	Package	Location in the X array of the first element of array DELR.
LCHCOF	Package	Location in the X array of the first element of array HCOF.
LCHNEW	Package	Location in the X array of the first element of array HNEW.
LCHOLD	Package	Location in the X array of the first element of array HOLD.
LCIBOU	Package	Location in the X array of the first element of array IBOUND.
LCIOFL	Package	Location in the X array of the first element of array IOFLG.
LCRHS	Package	Location in the X array of the first element of array RHS.
LCSTRT	Package	Location in the X array of the first element of array STRT.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRCL	Module	Number of cells in the grid.
NROW	Global	Number of rows in the grid.

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.

Code	Status
negative	constant head
zero	inactive (no-flow)
positive	variable head

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 SBAS1I to initialize the Output Control System.

9. RETURN.

Flow Chart for Module BAS1RP

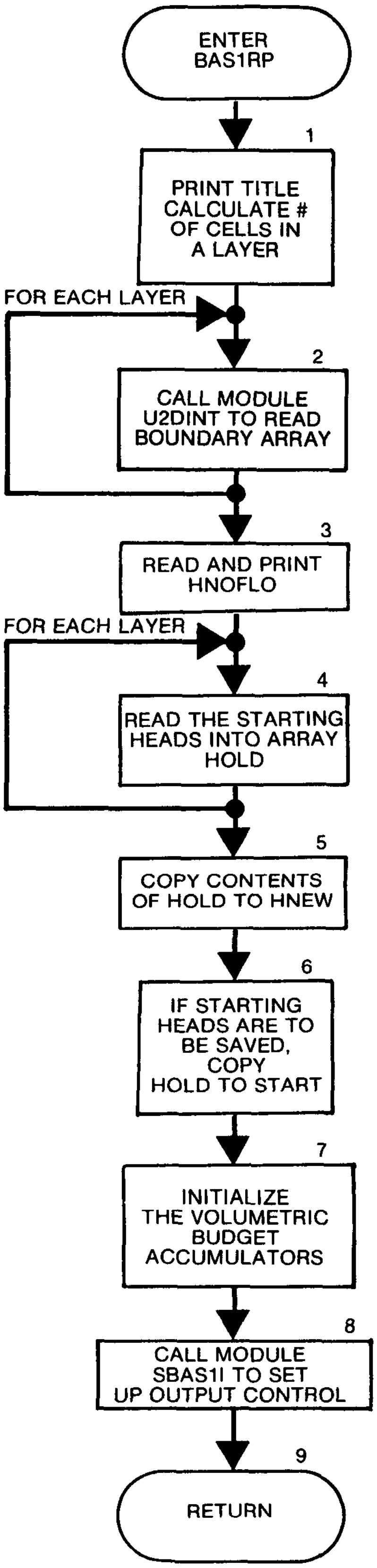
HNOFLO is a value assigned to head in inactive (no-flow) cells. It makes those cells stand out in listings of heads.

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.


ENTER BAS1RP
1
PRINT TITLE CALCULATE # OF CELLS IN A LAYER
FOR EACH LAYER
2
CALL MODULE U2DINT TO READ BOUNDARY ARRAY
3
READ AND PRINT HNOFLO
FOR EACH LAYER
4
READ THE STARTING HEADS INTO ARRAY HOLD
5
COPY CONTENTS OF HOLD TO HNEW
6
IF STARTING HEADS ARE TO BE SAVED, COPY HOLD TO START
7
INITIALIZE THE VOLUMETRIC BUDGET ACCUMULATORS
8
CALL MODULE SBAS1I TO SET UP OUTPUT CONTROL
9
RETURN

```
      SUBROUTINE BAS1RP(IBOUND,HNEW,STRT,HOLD,ISTRT,INBAS,
     1    HEADNG,NCOL,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,
     2    IDDNFM,IHEDUN,IDDNUN,IOUT)
C-----VERSION 1628 15MAY1987 BAS1RP
C     ******************************************************************
C     READ AND INITIALIZE BASIC MODEL ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 HEADNG,ANAME
      DOUBLE PRECISION HNEW,HNOFLO
C
      DIMENSION HNEW(NODES),IBOUND(NODES),STRT(NODES),HOLD(NODES),
     1             ANAME(6,2),VBVL(4,20),IOFLG(NLAY,4),HEADNG(32)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1   ANAME(6,1) /'    ','    ','  BO','UNDA','RY A','RRAY'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1   ANAME(6,2) /'    ','    ','    ','INIT','IAL ','HEAD'/
C     ------------------------------------------------------------------
C
C1------PRINT SIMULATION TITLE, CALCULATE # OF CELLS IN A LAYER.
      WRITE(IOUT,1) HEADNG
    1 FORMAT(1H1,32A4)
      NCR=NCOL*NROW
C
C2------READ BOUNDARY ARRAY(IBOUND) ONE LAYER AT A TIME.
      DO 100 K=1,NLAY
      KK=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(F10.0)
      HNOFLO=TMP
      WRITE(IOUT,3) TMP
    3 FORMAT(1H0,'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
      KK=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.0) HNEW(I)=HNOFLO
  400 CONTINUE
C
C6------IF STARTING HEADS ARE TO BE SAVED THEN COPY HOLD TO STRT.
      IF(ISTRT.EQ.0) 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,I)=0.
  600 CONTINUE
C
C8------SET UP OUTPUT CONTROL.
      CALL SBAS1I(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM,
     1         IDDNFM,IHEDUN,IDDNUN)
C
C9------RETURN
 1000 RETURN
      END
```

List of Variables for Module BAS1RP

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

Narrative for Module BAS1ST

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

DELT = (1-TSMULT)*PERLEN/(1-TSMULT**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.

Flow Chart for Module BAS1ST

PERLEN is the length of a stress period.

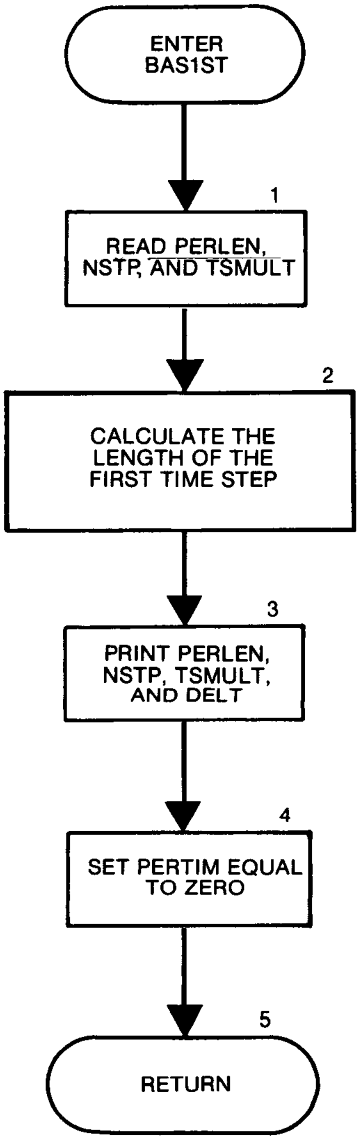
NSTP is the number of time steps in a stress period.

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=(1-TSMULT)*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.


ENTER BAS1ST
1
READ PERLEN, NSTP, AND TSMULT
2
CALCULATE THE LENGTH OF THE FIRST TIME STEP
3
PRINT PERLEN, NSTP, TSMULT, AND DELT
4
SET PERTIM EQUAL TO ZERO
5
RETURN

```
      SUBROUTINE BAS1ST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C
C-----VERSION 1614 08SEP1982 BAS1ST
C     ******************************************************************
C     SETUP TIME PARAMETERS FOR NEW TIME PERIOD
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
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(1H1,51X,'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 BAS1ST

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

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.

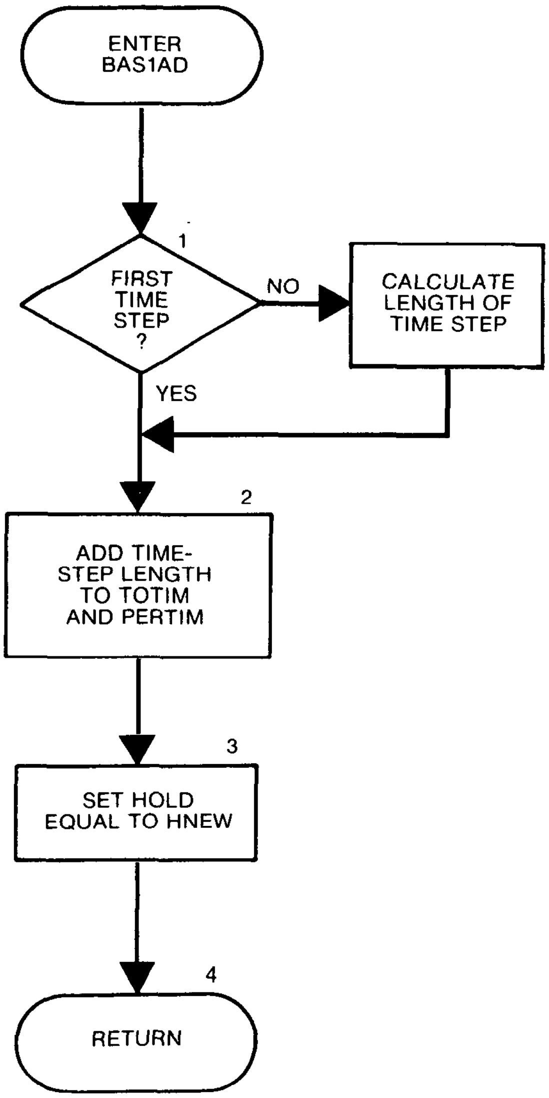
Flow Chart for Module BAS1AD

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.


ENTER BAS1AD
1
FIRST TIME STEP ?
NO
CALCULATE LENGTH OF TIME STEP
YES
2
ADD TIME-STEP LENGTH TO TOTIM AND PERTIM
3
SET HOLD EQUAL TO HNEW
4
RETURN

```
      SUBROUTINE BAS1AD(DELT,TSMULT,TOTIM,PERTIM,HNEW,HOLD,KSTP,
     1                  NCOL,NROW,NLAY)
C
C-----VERSION 1412 22FEB1982 BAS1AD
C
C     ******************************************************************
C     ADVANCE TO NEXT TIME STEP
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY), HOLD(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NOT FIRST TIME STEP THEN CALCULATE TIME STEP LENGTH.
      IF(KSTP.NE.1) DELT=TSMULT*DELT
C
C2------ACCUMULATE ELAPSED TIME IN SIMULATION(TOTIM) AND IN THIS
C2------STRESS PERIOD(PERTIM).
      TOTIM=TOTIM+DELT
      PERTIM=PERTIM+DELT
C
C3------COPY HNEW TO HOLD.
      DO 10 K=1,NLAY
      DO 10 I=1,NROW
      DO 10 J=1,NCOL
   10 HOLD(J,I,K)=HNEW(J,I,K)
C
C4------RETURN
      RETURN
      END
```

List of Variables for Module BAS1AD

Variable	Range	Definition
DELT	Global	Length of the current time step.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HOLD	Global	DIMENSION (NCOL,NROW,NLAY), Head at the start of the current time step.
I	Module	Row index.
J	Module	Column index.
K	Module	Layer index.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
PERTIM	Package	Elapsed time during the current stress period.
TOTIM	Package	Elapsed time in the simulation.
TSMULT	Package	Multiplier to get from one time step length to the next.

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 BAS1FM(HCOF,RHS,NODES)
C
C
C-----VERSION 1632 24JUL1987 BAS1FM
C     ******************************************************************
C     SET HCOF=RHS=0.
C     ******************************************************************
C
C        SPECIFICATIONS:
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

Variable	Range	Definition
I	Module	Index.
HCOF	Global	DIMENSION (NODES), Coefficient of head in cell (J,I,K) in the finite-difference equation.
NODES	Global	Number of cells (nodes) in the finite-difference grid.
RHS	Global	DIMENSION (NODES), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

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.

Flow Chart for Module BAS1OC

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:

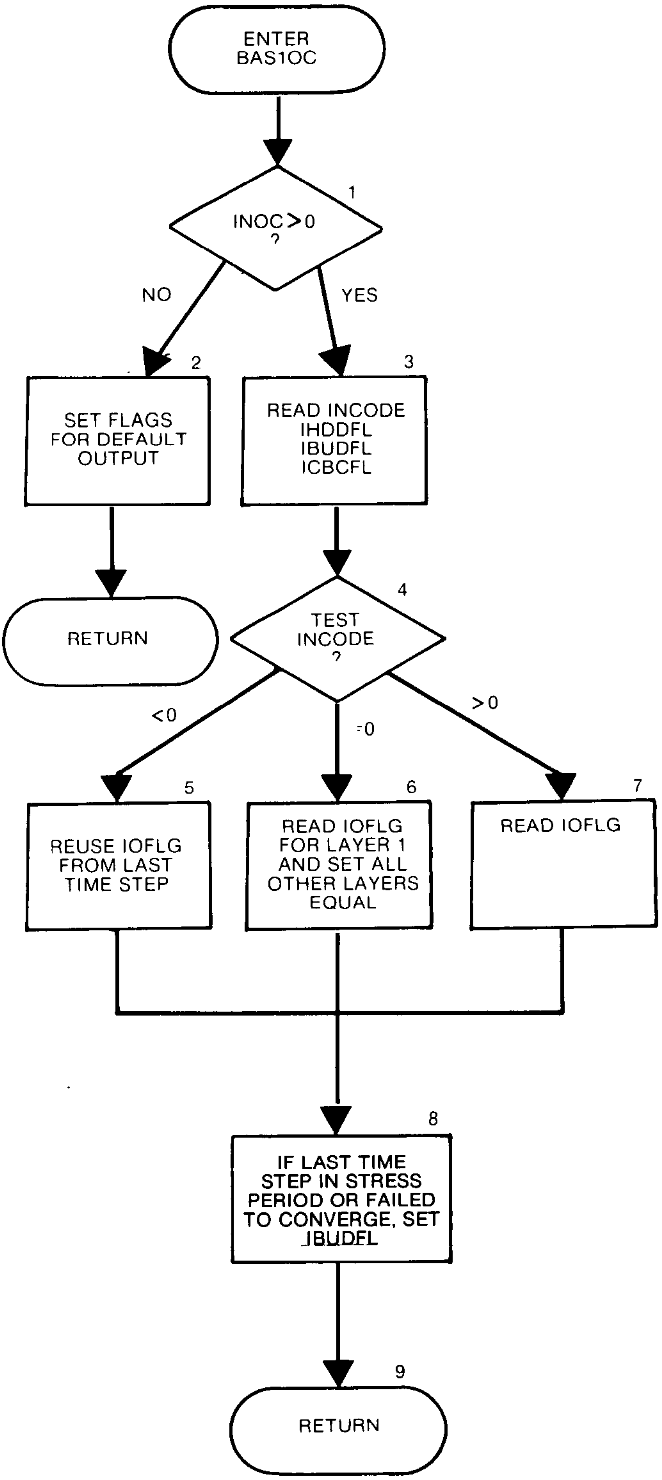
1---head print
2---drawdown print
3---head save
4---drawdown save

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.


ENTER BAS1OC
1
INOC>0 ?
NO
YES
2
SET FLAGS FOR DEFAULT OUTPUT
3
READ INCODE IHDDFL IBUDFL ICBCFL
RETURN
4
TEST INCODE ?
<0
=0
>0
5
REUSE IOFLG FROM LAST TIME STEP
6
READ IOFLG FOR LAYER 1 AND SET ALL OTHER LAYERS EQUAL
7
READ IOFLG
8
IF LAST TIME STEP IN STRESS PERIOD OR FAILED TO CONVERGE, SET IBUDFL
9
RETURN

```
      SUBROUTINE BAS1OC(NSTP,KSTP,ICNVG,IOFLG,NLAY,
     1      IBUDFL,ICBCFL,IHDDFL,INOC,IOUT)
C
C-----VERSION 1632 24JUL1987 BAS1OC
C     ******************************************************************
C     OUTPUT CONTROLLER FOR HEAD, DRAWDOWN, AND BUDGET
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION IOFLG(NLAY,4)
C     ------------------------------------------------------------------
C
C
C1------TEST UNIT NUMBER (INOC (INOC=IUNIT(12))) TO SEE IF
C1------OUTPUT CONTROL IS ACTIVE.
      IF(INOC.NE.0)GO TO 500
C
C2------IF OUTPUT CONTROL IS INACTIVE THEN SET DEFAULTS AND RETURN.
      IHDDFL=0
      IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP)IHDDFL=1
      IBUDFL=0
      IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP)IBUDFL=1
      ICBCFL=0
      GO TO 1000
C
C3-------READ AND PRINT OUTPUT FLAGS AND CODE FOR DEFINING IOFLG.
  500 READ(INOC,1) INCODE,IHDDFL,IBUDFL,ICBCFL
    1 FORMAT(4I10)
      WRITE(IOUT,3) IHDDFL,IBUDFL,ICBCFL
    3 FORMAT(1H0,'HEAD/DRAWDOWN PRINTOUT FLAG =',I2,
     1     5X,'TOTAL BUDGET PRINTOUT FLAG =',I2,
     2     5X,'CELL-BY-CELL FLOW TERM FLAG =',I2)
C
C4------DECODE INCODE TO DETERMINE HOW TO SET FLAGS IN IOFLG.
      IF(INCODE) 100,200,300
C
C5------USE IOFLG FROM LAST TIME STEP.
  100 WRITE(IOUT,101)
  101 FORMAT(1H ,'REUSING PREVIOUS VALUES OF IOFLG')
      GO TO 600
C
C6------READ IOFLG FOR LAYER 1 AND ASSIGN SAME TO ALL LAYERS
  200 READ(INOC,201) (IOFLG(1,M),M=1,4)
  201 FORMAT(4I10)
      DO 210 K=1,NLAY
      IOFLG(K,1)=IOFLG(1,1)
      IOFLG(K,2)=IOFLG(1,2)
      IOFLG(K,3)=IOFLG(1,3)
      IOFLG(K,4)=IOFLG(1,4)
  210 CONTINUE
      WRITE(IOUT,211) (IOFLG(1,M),M=1,4)
  211 FORMAT(1H0,'OUTPUT FLAGS FOR ALL LAYERS ARE THE SAME:'/
     1   1X,'  HEAD    DRAWDOWN  HEAD  DRAWDOWN'/
     2   1X,'PRINTOUT  PRINTOUT  SAVE    SAVE'/
     3   1X,34('-')/1X,I5,I10,I8,I8)
      GO TO 600
C
C7------READ IOFLG IN ENTIRETY
  300 READ(INOC,301) ((IOFLG(K,I),I=1,4),K=1,NLAY)
  301 FORMAT(4I10)
      WRITE(IOUT,302)
  302 FORMAT(1H0,'OUTPUT FLAGS FOR EACH LAYER:'/
     1   1X,'         HEAD    DRAWDOWN  HEAD  DRAWDOWN'/
     2   1X,'LAYER  PRINTOUT  PRINTOUT  SAVE    SAVE'/
     3   1X,41('-'))
      WRITE(IOUT,303) (K,(IOFLG(K,I),I=1,4),K=1,NLAY)
  303 FORMAT(1X,I4,I8,I10,I8,I8)
C
C8------THE LAST STEP IN A STRESS PERIOD AND STEPS WHERE ITERATIVE
C8------PROCEDURE FAILED TO CONVERGE GET A VOLUMETRIC BUDGET.
  600 IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP) IBUDFL=1
C
C9------RETURN
 1000 RETURN
      END
```

List of Variables for Module BAS1OC

Variable	Range	Definition
I	Module	Index.
IBUDFL	Package	Flag. = 0, volumetric budget will not be printed for the current time step. ≠ 0, volumetric budget should be printed for the current time step.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be recorded or printed for the current time step.
ICNVG	Global	Flag is set equal to 1 when the iteration procedure has converged.
IHDDFL	Package	Flag. = 0, neither head nor drawdown will be printed at this time step. ≠ 0, head and drawdown may be printed at the end of the current time step.
INCODE	Module	Code specified by user. < 0, reuse contents of IOFLG from the last time step. = 0, read IOFLG for layer 1 and set all other layers to the same thing. > 0, read IOFLG contents for each layer.
INOC	Package	Unit number from which input to output control option will be read.
IOFLG	Package	DIMENSION (NLAY,4), Flags to control printing and recording of head and drawdown for each layer. (NLAY,1) ≠ 0, heads will be printed. (NLAY,2) ≠ 0, drawdown will be printed. (NLAY,3) ≠ 0, heads will be recorded. (NLAY,4) ≠ 0, drawdown will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
K	Module	Layer index.
KSTP	Global	Time step counter. Reset at the start of each stress period.
M	Module	Index.
NLAY	Global	Number of layers in the grid.
NSTP	Global	Number of time steps in the current stress period.