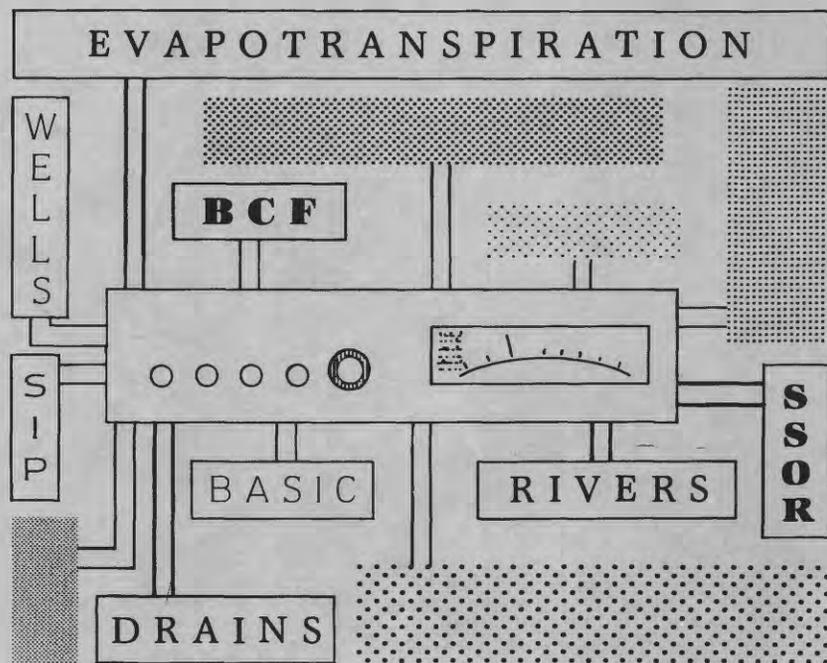


# A MODULAR THREE-DIMENSIONAL FINITE-DIFFERENCE GROUND-WATER FLOW MODEL



U.S. GEOLOGICAL SURVEY  
Open-File Report 83-875

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

by Michael G. McDonald and Arlen W. Harbaugh



U.S. GEOLOGICAL SURVEY  
Open-File Report 83—875  
1984

UNITED STATES DEPARTMENT OF THE INTERIOR  
WILLIAM P. CLARK, Secretary

GEOLOGICAL SURVEY  
Dallas L. Peck, Director

---

For additional information  
write to:

Chief Hydrologist  
U.S. Geological Survey, WRD  
411 National Center  
12201 Sunrise Valley Drive  
Reston, Virginia 22092

Copies of this report can  
be purchased from:

Open-File Services Section  
Western Distribution Branch  
Box 25425, Federal Center  
Denver, Colorado 80225  
Telephone: (303) 234-5888

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

CONTENTS

	Page
Abstract.....	1
Chapter 1. Introduction.....	2
Purpose.....	2
Organization of This Report.....	3
How to Use This Report.....	6
Chapter 2. Derivation of the Finite-Difference Equation.....	7
Mathematical Model.....	7
Discretization Convention.....	8
Finite-Difference Equation.....	12
Iteration.....	27
Chapter 3. Program Design.....	31
Boundaries.....	41
Space Allocation.....	42
Input Structure.....	42
Output Structure.....	47
Main Program.....	47
FORTRAN Listing of the Main Program.....	50
Chapter 4. Basic Package.....	53
Conceptualization and Implementation.....	53
Model Input and Selection of Major Options.....	53
Discretization of Space.....	54
Boundaries.....	60
Initial Conditions.....	60
Discretization of Time.....	63
Output.....	63
Input Instructions.....	67
Sample Input.....	71
Input Instructions for Output Control.....	72
Sample Input for Output Control.....	75
Module Documentation.....	76
BAS1DF.....	77
BAS1AL.....	81
BAS1RP.....	85
BAS1ST.....	89
BAS1AD.....	93
BAS1FM.....	97
BAS1OC.....	100
BAS1OT.....	104
SBAS1D.....	109
SBAS1H.....	113
SABS1I.....	117
SBAS1T.....	121
SBAS1V.....	124
Chapter 5. Block-Centered Flow Package.....	129
Conceptualization and Implementation.....	129
Basic Conductance Equations.....	130
Horizontal Conductance.....	134

	Page
Vertical Conductance.....	138
A Variation of the Vertical Conductance Formulation.....	144
Storage Terms.....	147
Limitations on Use.....	149
Data Requirements.....	149
Input Instructions.....	153
Sample Input.....	156
Module Documentation.....	157
BCF1AL.....	159
BCF1RP.....	165
BCF1FM.....	171
BCF1BD.....	177
SBCF1N.....	183
SBCF1H.....	188
SBCF1C.....	192
SBCF1B.....	196
SBCF1F.....	201
Chapter 6. River Package.....	209
Conceptualization and Implementation.....	209
Input Instructions.....	218
Sample Input.....	220
Module Documentation.....	221
RIV1AL.....	222
RIV1RP.....	226
RIV1FM.....	230
RIV1BD.....	234
Chapter 7. Recharge Package.....	241
Conceptualization and Implementation.....	241
Input Instructions.....	244
Sample Input.....	246
Module Documentation.....	247
RCH1AL.....	248
RCH1RP.....	252
RCH1FM.....	256
RCH1BD.....	260
Chapter 8. Well Package.....	267
Conceptualization and Implementation.....	267
Input Instructions.....	268
Sample Input.....	269
Module Documentation.....	270
WEL1AL.....	271
WEL1RP.....	275
WEL1FM.....	279
WEL1BD.....	282
Chapter 9. Drain Package.....	288
Conceptualization and Implementation.....	288
Input Instructions.....	293
Sample Input.....	295
Module Documentation.....	296
DRN1AL.....	297
DRN1RP.....	301
DRN1FM.....	305
DRN1BD.....	309

	Page
Chapter 10. Evapotranspiration Package.....	316
Conceptualization and Implementation.....	316
Input Instructions.....	321
Sample Input.....	323
Module Documentation.....	324
EVT1AL.....	325
EVT1RP.....	329
EVT1FM.....	333
EVT1BD.....	337
Chapter 11. General-Head Boundary Package.....	343
Conceptualization and Implementation.....	343
Input Instructions.....	347
Sample Input.....	349
Module Documentation.....	350
GHB1AL.....	351
GHB1RP.....	355
GHB1FM.....	359
GHB1BD.....	363
Chapter 12. Strongly Implicit Procedure Package.....	370
Conceptualization and Implementation.....	370
Input Instructions.....	396
Sample Input.....	397
Module Documentation.....	398
SIP1AL.....	399
SIP1RP.....	403
SIP1AP.....	407
SSIP1P.....	422
SSIP1I.....	425
Chapter 13. Slice-Successive Overrelaxation Package.....	432
Conceptualization and Implementation.....	432
Input Instructions.....	439
Module Documentation.....	440
SOR1AL.....	441
SOR1RP.....	445
SOR1AP.....	448
SSOR1B.....	458
Chapter 14. Utility Modules.....	464
Input Instructions for Array Readers.....	467
UBUDSV.....	469
ULASAV.....	472
ULAPRS.....	475
ULAPRW.....	480
UCOLNO.....	485
U2DREL.....	489
U2DINT.....	493
U1DREL.....	498
References.....	502
Appendix A--Program Portability.....	503
Appendix B--Space Requirements in the X Array.....	506
Appendix C--Continuation of a Previous Run.....	508
Appendix D--Sample Problem.....	509
Appendix E--Abbreviated Input Instructions.....	521

## ILLUSTRATIONS

	Page
Figure 1. A discretized hypothetical aquifer system-----	9
2. Grids showing the difference between block-centered and point-centered grids-----	11
3. Cell i,j,k and indices for the six adjacent cells-----	14
4. Flow into cell i,j,k from cell i,j-1,k-----	15
5. Leakage through a riverbed into a cell-----	19
6. Hydrograph for cell i,j,k-----	22
7. Discretized aquifer showing boundaries and constant-head cells-----	25
8. Iterative calculation of a head distribution-----	28
9. Overall program structure-----	32
10. Organization of modules by procedures and packages-----	37
11. Primary modules organized by procedure and package-----	39
12. Overall program structure showing all primary modules----	40
13. Specification of major options using the IUNIT array-----	44
14. Sample job showing role of the IUNIT array-----	45
15. Effect of using distorted grid in the vertical direction-----	55
16. Flow net in a cross section consisting of two high conductivity units separated by a low conductivity unit-----	56
17. A cross section in which a low conductivity unit is represented by six model layers-----	57
18. A cross section in which a low conductivity unit is not represented by a model layer-----	59
19. Example of the boundary array (IBOUND) for a single layer-----	61
20. Flow of head distributions during a simulation-----	62
21. Division of simulation time into stress periods and time steps-----	64
22. Sample overall volumetric water budget-----	65
23. Prism of porous material to which Darcy's law may be applied-----	131
24. Calculation of conductance through several prisms in series-----	133
25. Calculation of conductance between nodes using transmissivity and dimensions of cells-----	135
26. Calculating vertical conductance between adjacent cells-----	139
27. Using a grid, distorted in the vertical direction, to represent lithologic units of varying thickness-----	141
28. Calculating vertical conductance between two cells which are separated by a confining unit-----	143
29. Situation in which flow down into an unconfined cell may be overstated-----	146
30. A model cell which uses two storage factors during one iteration-----	146
31. Relationship among the modules in the Block-Centered Flow Package-----	158
32. Water table contours near (a) a gaining stream and (b) a losing stream-----	210
33. Discretization of a river into reaches. Some small reaches are ignored-----	211

Figure 34.	(a) Cross section of an aquifer containing a river and (b) block diagram of a cross section of an aquifer containing a river-----	212
35.	A riverbed viewed as a prism of porous material-----	213
36.	Cross section showing the relation between head on the aquifer side of the riverbed and head in the cell. Head in the cell is equal to the water-table elevation-----	215
37.	Leakage through a riverbed into an aquifer as a function of head in the aquifer-----	217
38.	Hypothetical problem showing which cells receive recharge under the three options available in the Recharge Package-----	243
39.	Flow into a drain as a function of head in the aquifer and the elevation of the drain-----	289
40.	Factors that may affect the conductance of the interface between an aquifer and a drain-----	290
41.	Leakage into a drain as a function of head in the aquifer-----	291
42.	Evapotranspiration as a function of head in the aquifer-----	317
43.	Hypothetical problem showing from which cells ET will be abstracted under the two options available in the ET Package-----	320
44.	Three situations which can be simulated using the General-Head Boundary Package: (a) a gaining stream, (b) a buried drain, and (c) horizontal leakage-----	344
45.	Flow from a general-head boundary as a function of head in the aquifer-----	345
46.	Correspondence between the finite-difference equations and the matrix equation for a grid of three rows, four columns, and two layers-----	372
47.	Structure of coefficient matrix showing nonzero diagonals-----	373
48.	Symmetric coefficient matrix for a grid containing two rows, three columns, and two layers-----	374
49.	Decomposition of a coefficient matrix into lower and upper triangular matrices-----	375
50.	Desired structure, showing nonzero diagonals, of (a) the lower triangular factor $\underline{L}$ of $\underline{A+B}$ , and (b) the upper triangular factor $\underline{U}$ of $\underline{A+B}$ -----	378
51.	Structure of matrix $\underline{A+B}$ showing nonzero diagonals-----	379
52.	Estimation of a function at one corner of a rectangle in terms of the function at the other three corners----	382
53.	Cell numbering schemes for a grid using three indices and using one index-----	386
54.	SSOR reduces the number of equations that must be solved simultaneously by considering a single vertical slice at a time-----	434
55.	Organization of elements in a matrix for a slice in conventional notation and in compressed notation-----	437
56.	Illustration of wrap and strip forms of printed output for a layer containing 7 rows and 17 columns----	465

## TABLES

	Page
Table 1. List of packages-----	35
2. Print-format codes for utility modules ULAPRS and ULAPRW-----	466

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

By Michael G. McDonald and Arlen W. Harbaugh

## ABSTRACT

This report presents a finite-difference model and its associated modular computer program. The model simulates flow in three dimensions. The report includes detailed explanations of physical and mathematical concepts on which the model is based and an explanation of how those concepts were incorporated in the modular structure of the computer program. The modular structure consists of a Main Program and a series of highly independent subroutines called "modules." The modules are grouped into "packages." Each package deals with a specific feature of the hydrologic system which is to be simulated, such as flow from rivers or flow into drains, or with a specific method of solving linear equations which describe the flow system, such as the Strongly Implicit Procedure or Slice-Successive Overrelaxation.

The division of the program into modules permits the user to examine specific hydrologic features of the model independently. This also facilitates development of additional capabilities because new modules or packages can be added to the program without modifying the existing modules or packages. The input and output systems of the computer program are also designed to permit maximum flexibility.

Ground-water flow within the aquifer is simulated using a block-centered finite-difference approach. Layers can be simulated as confined, unconfined, or a combination of confined and unconfined. Flow from external stresses, such as flow to wells, areal recharge, evapotranspiration, flow to drains, and flow through riverbeds, can also be simulated. The finite-difference equations can be solved using either the Strongly Implicit Procedure or Slice-Successive Overrelaxation.

The program is written in FORTRAN '66 and will run without modification on most computers which have a FORTRAN '66 compiler. It will also run, without modification, with most extended FORTRAN '77 compilers and with minor modifications on standard FORTRAN '77 compilers. Documentation presented in this report includes a narrative description, a flow chart, a list of variables, and a program listing for each module.

## CHAPTER 1

### INTRODUCTION

#### Purpose

Since their inception, the two- and three-dimensional finite-difference models of Trescott (1975), and Trescott, Pinder, and Larson (1976) have been used extensively by the U.S. Geological Survey and others for the computer simulation of ground-water flow. In many cases, users of these models have found it necessary to add various options to the original programs or to modify the programs for assorted reasons. The design of these prototype models is such that, in most cases, adding options or capabilities require alterations throughout the original program. The result has been the creation of a conglomeration of models, each differing in varying degrees from the original programs. The main objectives in designing a new ground-water flow model were to produce a program that can be readily modified, is simple to use and maintain, can be executed on a variety of computers with minimal changes, and is relatively efficient with respect to computer memory and execution time.

The model program documented in this report uses a modular programming structure wherein similar programming functions are grouped together and specific computational and hydrologic options are constructed in such a manner that each option is independent of other options. Because of this structure, new options can be added without the necessity of changing existing subroutines. In addition, subroutines pertaining to options that are not being used can be deleted, thereby reducing the size of the program. The model may be used for either two- or three-dimensional applications. Input procedures have been generalized so that each type of model input

data may be stored and read from separate external files. Variable formatting allows input data arrays to be read in any format without modification to the program. The type of output that is available has also been generalized so that the user may select various model output options to suit a particular need. The program, which is written in FORTRAN '66, has been successfully run without modification on computers manufactured by IBM, Control Data, Prime, Amdahl, Digital Equipment, and Cray corporations.<sup>1/</sup>

The major options that are presently available include procedures to simulate the effects of wells, recharge, rivers, drains, evapotranspiration, and general-head boundaries. The solution algorithms available include two iteration techniques, the Strongly Implicit Procedure (SIP) and the Slice-Successive Overrelaxation method (SSOR).

#### Organization of This Report

The purpose of this report is to describe the mathematical concepts used in this program, the design of the program, and the input needed to use the program. The program has been divided into a main program and a series of highly independent subroutines called modules. The modules, in turn, have been grouped into "packages." A package is a group of modules that deals with a single aspect of the simulation. For example, the Well Package simulates the effect of wells, the River Package simulates the effect of rivers, and the SIP Package solves a system of equations using the Strongly Implicit Procedure. Most of the packages are options which the user may

<sup>1</sup>"Use of IBM, Control Data, Prime, Amdahl, Digital Equipment, and Cray corporations in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey."

or may not have occasion to use. Each package is described in a separate chapter so that the user has to read only about those packages that he intends to use. Two preliminary chapters describe topics relating to the overall program; chapter 2 derives the finite-difference equation that is used in all of the other chapters and chapter 3 describes the overall design of the program. Chapter 14 describes utility modules used by the packages. Appendices A-E cover topics relating to the operation of the model.

Chapters 4 through 13 describe individual packages. Description of a package consists of (1) a section entitled "Conceptualization and Implementation," (2) input instructions for the package, (3) sample input, and (4) module documentation. The Conceptualization and Implementation section describes the physical and mathematical concepts used to build the package. For example, the package that describes the River Package derives an equation which approximates flow through a riverbed and shows how that equation can be incorporated into the finite-difference equation. The chapter that describes the Strongly Implicit Procedure explains a method for solving a system of linear equations.

Input instructions describe "items" of input. An item is a single record or a collection of similar records, or an array or a collection of similar arrays. Each item is numbered. The description of an item which is a record or group of records contains a line which names the fields contained in the records and a line showing the format of the fields. The format is given in standard FORTRAN. The description of an item which is an array or a group of arrays contains a line that names the array and a line that names the utility module which reads the array. Details about the utility modules are contained in chapter 14. Immediately following the list of

items is a list of definitions of input fields and arrays. Fields or arrays which are variables in the program are capitalized; fields or arrays that are used only in the input instructions are in both upper- and lower-case letters. A line immediately after an item tells whether it consists of more than one record or more than one array. Input to each package is read from a unit number specified by the user in an array named "IUNIT" (see an explanation of the IUNIT array in the chapter covering the Basic Package). The element in the IUNIT array that corresponds to a particular package is listed at the top of the input instructions. The name of the module which reads each input item is printed in the center of the page immediately before the first item read by the module.

Utility modules are subroutines which perform tasks for several different packages. For example, modules U2DREL, U2DINT, and U1DREL read arrays of values for various packages. Modules ULAPRW and ULAPRS print arrays of values. The utility modules are described in chapter 14.

Module documentation consists of a list of modules in the package and detailed descriptions of each of the modules. The detailed description of a module contains four documents: (1) a narrative description of the module, (2) a flow chart of the module, (3) a FORTRAN listing of the module, and (4) a list of the variable names which are used in the module. For very simple modules, the flow chart is omitted. The narrative description is a numbered list of the functions performed by the module showing the order in which they are performed. The flow chart is a graphic equivalent of the narrative. The blocks in the flow chart are numbered with the same numbers used in the narrative so that the two documents can be cross referenced. An explanation of terms used in the flow chart is contained

on the sheet with the flow chart. The program listing contains comments with numbers corresponding to those used in the flow charts and the narratives. The fourth record of the listing contains a comment showing the time and day that the module was last modified. The list of variables shows the name, range, and definition of every variable used in the module. The range indicates if the variable is used in only one module--"Module," if it is used in only one package--"Package," or if it is used in more than one package--"Global."

### How to Use This Report

To understand the overall design of this program, read chapters 2 and 3 and "Conceptualization and Implementation" in chapter 4. To understand the formulation of coefficients representing flow within the aquifer, read "Conceptualization and Implementation" in chapter 5. To understand how a particular external source or sink is represented, read the "Conceptualization and Implementation" section of the corresponding chapter (chapters 6 through 11). To understand how a particular solver works, read "Conceptualization and Implementation" in chapters 12 or 13. To run the program, read the input instructions for the appropriate packages and read about the utility modules in chapter 14. To get a deeper understanding of a particular facet of the program or to modify the program, identify and study the detailed description of the relevant modules. See the appendices for a sample problem, abbreviated input instructions, and computer-related considerations.

## CHAPTER 2

### DERIVATION OF THE FINITE-DIFFERENCE EQUATION

#### Mathematical Model

The three-dimensional movement of ground water of constant density through porous earth material may be described by the partial-differential equation

$$\frac{\partial}{\partial x} \left( K_{xx} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_{yy} \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_{zz} \frac{\partial h}{\partial z} \right) - W = S_s \frac{\partial h}{\partial t} \quad (1)$$

where

$x$ ,  $y$ , and  $z$  are cartesian coordinates aligned along the major axes of hydraulic conductivity  $K_{xx}, K_{yy}, K_{zz}$ ;

$h$  is the potentiometric head (L);

$W$  is a volumetric flux per unit volume and represents sources and/or sinks of water ( $t^{-1}$ );

$S_s$  is the specific storage of the porous material ( $L^{-1}$ ); and

$t$  is time (t).

In general,  $S_s, K_{xx}, K_{yy}, K_{zz}$  may be functions of space ( $S_s = S_s(x, y, z)$ , and  $K_{xx} = K_{xx}(x, y, z)$ , etc.) and  $h$  and  $W$  may be functions of space and time ( $h = h(x, y, z, t)$ ,  $W = W(x, y, z, t)$ ) so that equation 1 describes ground-water flow under nonequilibrium conditions in a heterogeneous and anisotropic medium.

Equation 1, together with specification of flow and/or head conditions at the boundaries of an aquifer system and specification of initial-head conditions, constitutes a mathematical model of ground-water flow. A solution of equation 1, in an analytical sense, is an algebraic expression giving  $h(x, y, z, t)$  such that, when the derivatives of  $h$  with respect to

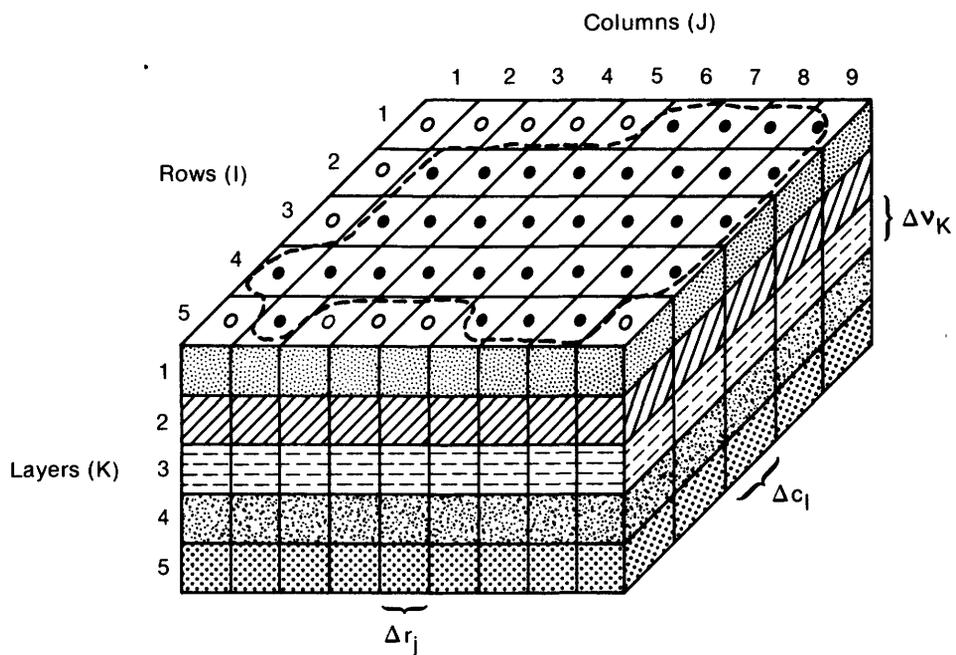
space and time are substituted into equation 1, the equation and its initial and boundary conditions are satisfied. A time-varying head distribution of this nature characterizes the flow system in that it measures both the energy of flow and the volume of water in storage and can be used to calculate directions and rates of movement.

Except for very simple systems, analytical solutions of equation 1 are rarely possible so various numerical methods must be employed to obtain approximate solutions. One such approach is the finite-difference method wherein the continuous system described by equation 1 is replaced by a finite set of discrete points in space and time, and the partial derivatives are replaced by differences between functional values at these points. The process leads to systems of simultaneous linear algebraic difference equations; their solution yields values of head at specific points and time. These values constitute an approximation to the time-varying head distribution that would be given by an analytical solution of the partial-differential equation of flow.

The finite-difference analog of equation 1 may be derived by applying the rules of difference calculus; however, in the discussion presented here, an alternative approach is used with the aim of simplifying the mathematical treatment and explaining the computational procedure in terms of familiar physical concepts regarding the flow system.

#### Discretization Convention

Figure 1 shows a spatial discretization of an aquifer system into a mesh of points termed nodes, forming rows, columns, and layers. To conform with computer array conventions, an  $i,j,k$  coordinate system is used. For a



Explanation

----- Aquifer Boundary

● Active Cell

○ Inactive Cell

$\Delta r_j$  Dimension of Cell Along the Row Direction. Subscript (J) Indicates the Number of the Column

$\Delta c_l$  Dimension of Cell Along the Column Direction. Subscript (I) Indicates the Number of the Row

$\Delta v_k$  Dimension of the Cell Along the Vertical Direction. Subscript (K) Indicates the Number of the Layer

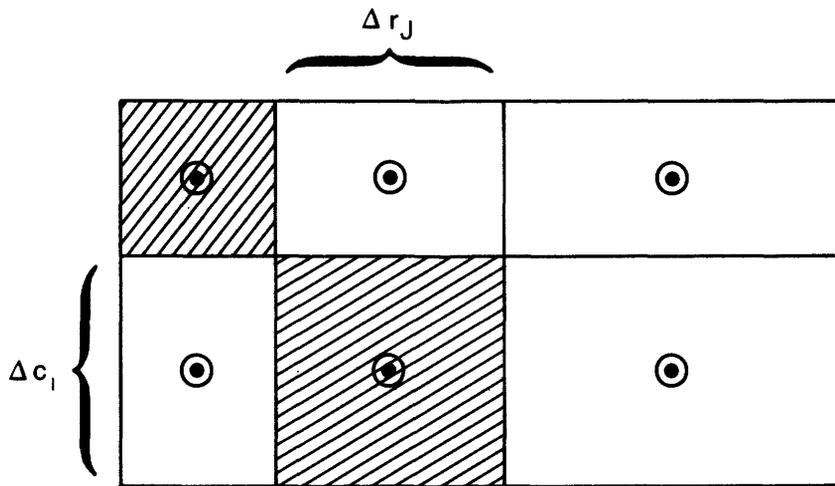
Figure 1.—A discretized hypothetical aquifer system.

system consisting of "nrow" rows, "ncol" columns, and "nlay" layers,  $i$  is the row index,  $i = 1, 2, \dots, nrow$ ;  $j$  is the column index,  $j = 1, 2, \dots, ncol$ ; and  $k$  is the layer index,  $k = 1, 2, \dots, nlay$ . For example, figure 1 shows a system with  $nrow = 5$ ,  $ncol = 9$ , and  $nlay = 5$ . The origin of the system  $(1, 1, 1)$ , is the upper-left corner of the topmost layer. With respect to a cartesian coordinate system, points along a row are parallel to the  $x$  axis, points along a column are parallel to the  $y$  axis, and points along the vertical are parallel to the  $z$  axis.

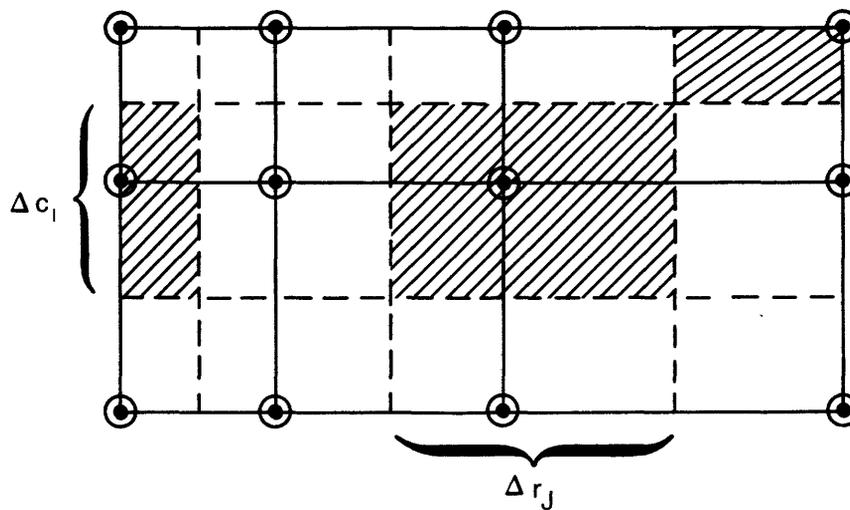
Conceptually, nodes represent prisms of porous material, termed cells, within which the hydraulic properties are constant so that any value associated with a node applies to or is distributed over the extent of a cell.

In figure 1, the width of cells along rows is designated as  $\Delta r_j$  for the  $j$ th column; the width of cells along columns are designated as  $\Delta c_i$  for the  $i$ th row; and the thickness of layers in the vertical are designated as  $\Delta v_k$  for the  $k$ th layer. Thus, the cell with coordinates of  $(i, j, k) = (4, 8, 3)$  has a volume of  $\Delta r_8 \Delta c_4 \Delta v_3$ .

Figure 2 shows two conventions for defining the configuration of cells with respect to the location of nodes--the block-centered formulation and the point-centered formulation. Both systems start by dividing the aquifer with two sets of parallel lines which are perpendicular to each other. In the block-centered formulation, the blocks formed by the sets of parallel lines are the cells; the nodes are at the center of the cells. In the point-centered formulation, the nodes are at the intersection points of the sets of parallel lines, and cells are drawn around the nodes with faces halfway between nodes. In either case, spacing of nodes should be such that the



Block-Centered Grid System



Point-Centered Grid System

Explanation

-  Nodes
-  Grid Lines
-  Cell Boundaries for Point Centered Formulation
-  Cells Associated With Selected Nodes

Figure 2.—Grids showing the difference between block-centered and point-centered grids.

hydraulic properties of the system are, in fact, uniform over the extent of a cell. The following development of the finite-difference equation holds for either formulation. Although the model can accept both formulations, only the block-centered formulation is included in this release.

In equation 1, the head  $h$  is a function of time as well as space so that, in the finite-difference formulation, discretization of the continuous time domain is required.

### Finite-Difference Equation

Development of the ground-water flow equation in finite-difference form follows from the application of the continuity equation: the sum of all flows into and out of the cell must be equal to the rate of change in storage within the cell. Under the assumption that the density of ground water is constant, the continuity equation expressing the balance of flow for a cell is

$$\sum Q_j = S_s \frac{\Delta h}{\Delta t} \Delta V \quad (2)$$

where

$Q_j$  is a flow rate into the cell ( $L^3t^{-1}$ );

$S_s$  is the specific storage defined as the ratio of the volume of water which can be injected per unit volume of aquifer material per unit change in head ( $L^{-1}$ );

$\Delta V$  is the volume of the cell ( $L^3$ ); and

$\Delta h$  is the change in head over a time interval of length  $\Delta t$ .

The term on the right hand side is equivalent to the volume of water taken into storage over a time interval  $\Delta t$  given a change in head of  $\Delta h$ .

Equation 2 is stated in terms of inflow and storage gain. Outflow and loss are represented by defining outflow as negative inflow and loss as negative gain.

Figure 3 depicts a cell  $i,j,k$  and six adjacent aquifer cells  $i-1,j,k$ ;  $i+1,j,k$ ;  $i,j-1,k$ ;  $i,j+1,k$ ;  $i,j,k-1$ ; and  $i,j,k+1$ . Flow into cell  $i,j,k$  in the row direction from cell  $i,j-1,k$  (fig. 3), according to Darcy's law, is given by

$$q_{i,j-1/2,k} = KR_{i,j-1/2,k} \Delta C_i \Delta V_k \frac{(h_{i,j-1,k} - h_{i,j,k})}{\Delta r_{j-1/2}} \quad (3)$$

where

$q_{i,j-1/2,k}$  is the volumetric fluid discharge through the face between cells  $i,j,k$  and  $i,j-1,k$  ( $L^3t^{-1}$ );

$KR_{i,j-1/2,k}$  is the hydraulic conductivity along the row between nodes  $i,j,k$  and  $i,j-1,k$  ( $Lt^{-1}$ ); and

$\Delta r_{j-1/2}$  is the distance between nodes  $i,j,k$  and  $i,j-1,k$  (L).

The index  $j-1/2$  is used to indicate the space between nodes (fig. 4). It does not indicate a point exactly halfway between nodes. For example,  $KR_{i,j-1/2,k}$  represents hydraulic conductivity in the entire region between nodes  $i,j,k$  and  $i,j-1,k$ .

Similar expressions can be written approximating the flow into or out of the cell through the remaining five faces, i.e., for flow in the row direction through the face between cells  $i,j,k$  and  $i,j+1,k$ ,

$$q_{i,j+1/2,k} = KR_{i,j+1/2,k} \Delta C_i \Delta V_k \frac{(h_{i,j+1,k} - h_{i,j,k})}{\Delta r_{j+1/2}} \quad (4)$$

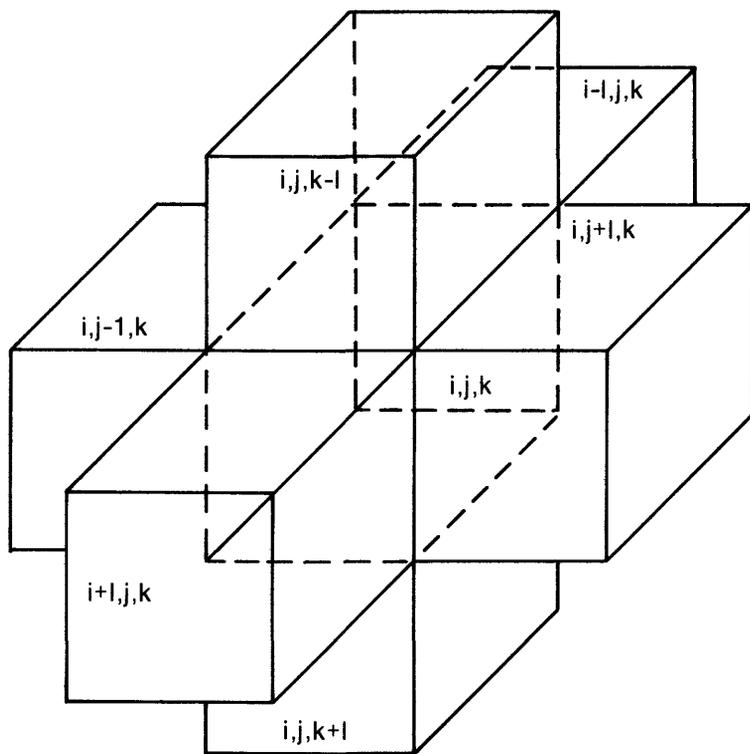


Figure 3.—Cell  $i,j,k$  and indices for the six adjacent cells.

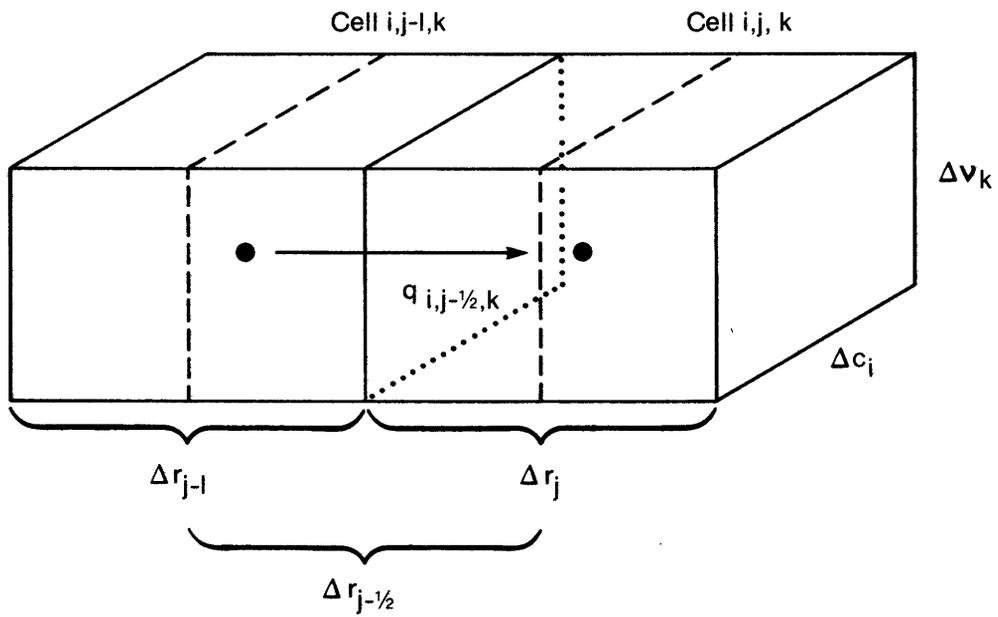


Figure 4.—Flow into cell  $i, j, k$  from cell  $i, j-1, k$ .

while for the column direction, flow through the forward face of the block is

$$q_{i+1/2,j,k} = KC_{i+1/2,j,k} \Delta r_j \Delta v_k \frac{(h_{i+1,j,k} - h_{i,j,k})}{\Delta c_{i+1/2}} \quad (5)$$

and flow through the rear face of the block is

$$q_{i-1/2,j,k} = KC_{i-1/2,j,k} \Delta r_j \Delta v_k \frac{(h_{i-1,j,k} - h_{i,j,k})}{\Delta c_{i-1/2}}. \quad (6)$$

For the vertical direction, flow through the bottom face is

$$q_{i,j,k+1/2} = KV_{i,j,k+1/2} \Delta r_j \Delta c_i \frac{(h_{i,j,k+1} - h_{i,j,k})}{\Delta v_{k+1/2}} \quad (7)$$

while flow through the upper face is given by

$$q_{i,j,k-1/2} = KV_{i,j,k-1/2} \Delta r_j \Delta c_i \frac{(h_{i,j,k-1} - h_{i,j,k})}{\Delta v_{k-1/2}}. \quad (8)$$

Each of equations 3-8 expresses flow through a face of cell  $i,j,k$  in terms of heads, grid dimensions, and hydraulic conductivity. Grid dimensions and hydraulic conductivity remain constant throughout the solution process so that the notation can be simplified by combining the constants into a single constant, which multiplies head, called the "hydraulic conductance" or, more simply, the "conductance." For example,

$$CR_{i,j-1/2,k} = KR_{i,j-1/2,k} \Delta c_i \Delta v_k / \Delta r_{j-1/2} \quad (9)$$

where

$CR_{i,j-1/2,k}$  is the conductance in row  $i$  and layer  $k$  between nodes  $i,j-1,k$  and  $i,j,k$  ( $L^2t^{-1}$ ).

Conductance is the product of hydraulic conductivity and cross-sectional area of flow divided by the length of the flow path; in this case, the distance between the nodes.

Substituting this expression into equation 3 yields

$$q_{i,j-1/2,k} = CR_{i,j-1/2,k}(h_{i,j-1,k} - h_{i,j,k}). \quad (10)$$

Similarly, equations 4-8 can be rewritten to yield

$$q_{i,j+1/2,k} = CR_{i,j+1/2,k}(h_{i,j+1,k} - h_{i,j,k}) \quad (11)$$

$$q_{i-1/2,j,k} = CC_{i-1/2,j,k}(h_{i-1,j,k} - h_{i,j,k}) \quad (12)$$

$$q_{i+1/2,j,k} = CC_{i+1/2,j,k}(h_{i+1,j,k} - h_{i,j,k}) \quad (13)$$

$$q_{i,j,k-1/2} = CV_{i,j,k-1/2}(h_{i,j,k-1} - h_{i,j,k}) \quad (14)$$

$$q_{i,j,k+1/2} = CV_{i,j,k+1/2}(h_{i,j,k+1} - h_{i,j,k}) \quad (15)$$

where conductances are defined analogously to  $CR_{i,j-1/2,k}$  in equation 9.

Equations 10-15 account for the flow into cell  $i,j,k$  from the six adjacent cells. To account for flows into the cell from outside the aquifer, such as seepage through streambeds, drains, areal recharge, evapotranspiration, and wells, additional terms are required. These flows may be dependent on the head in the receiving cell but independent of all other heads in the aquifer or they may be entirely independent of head in the receiving cell. Flow from outside the aquifer may be represented by the expression

$$a_{i,j,k,n} = p_{i,j,k,n}h_{i,j,k} + q_{i,j,k,n} \quad (16)$$

where

$a_{i,j,k,n}$  represents flow from the  $n$ -th external source into cell  $i,j,k$  ( $L^3t^{-1}$ ), and  $p_{i,j,k,n}$  and  $q_{i,j,k,n}$  are constants ( $L^2t^{-1}$  and  $L^3t^{-1}$ , respectively).

For example, suppose a cell is receiving flow from two sources, recharge from a well and seepage through a riverbed. For the first source ( $n = 1$ ), since the flow from the well is assumed to be independent of head,  $p_{i,j,k,1}$  is zero and  $q_{i,j,k,1}$  is the recharge rate for the well. In this case,

$$a_{i,j,k,1} = q_{i,j,k,1} \quad (17)$$

For the second source ( $n = 2$ ), the seepage is proportional to the head difference between river stage and head in the cell  $i,j,k$  (fig. 5) so that

$$a_{i,j,k,2} = CRIV_{i,j,k,2}(R_{i,j,k} - h_{i,j,k}) \quad (18)$$

where

$CRIV_{i,j,k,2}$  is the conductance of the riverbed (fig. 5) in cell  $i,j,k$  ( $L^2t^{-1}$ ), and  $R_{i,j,k}$  is the head in the river ( $L$ ).

Equation 18 can be rewritten as

$$a_{i,j,k,2} = - CRIV_{i,j,k,2}h_{i,j,k} + CRIV_{i,j,k,2}R_{i,j,k} \quad (19)$$

The conductance  $CRIV_{i,j,k,2}$  corresponds to  $p_{i,j,k,2}$  and the term  $CRIV_{i,j,k,2}R_{i,j,k}$  corresponds to  $q_{i,j,k,2}$ . Similarly, all other external sources or stresses can be represented by an expression of the form of equation 16. In general, if there are  $N$  external sources or stresses affecting a single cell, the combined flow is expressed by

$$Q_{S_{i,j,k}} = \sum_{n=1}^N a_{i,j,k,n} = \sum_{n=1}^N p_{i,j,k,n} h_{i,j,k} + \sum_{n=1}^N q_{i,j,k,n} \quad (20)$$

Defining  $P_{i,j,k}$  and  $Q_{i,j,k}$  by the expressions

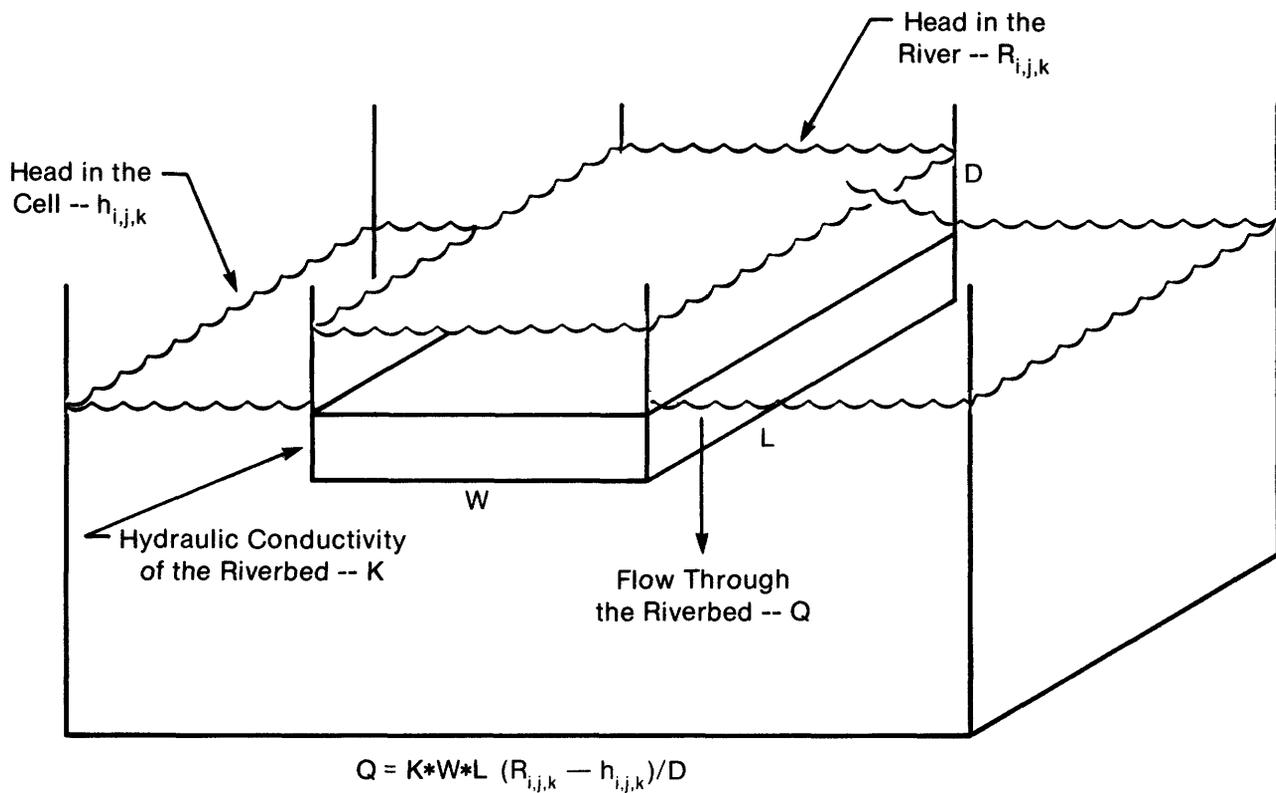


Figure 5.—Leakage through a riverbed into a cell.

$$P_{i,j,k} = \sum_{n=1}^N p_{i,j,k,n} \text{ and}$$

$$Q_{i,j,k} = \sum_{n=1}^N q_{i,j,k,n},$$

the general external flow term for cell  $i,j,k$  is

$$QS_{i,j,k} = P_{i,j,k} h_{i,j,k} + Q_{i,j,k}. \quad (21)$$

The continuity equation 2 including the flow rates between node  $i,j,k$ , the six adjacent nodes, and the external flow rate  $QS$  yields

$$\begin{aligned} & q_{i,j-1/2,k} + q_{i,j+1/2,k} + q_{i-1/2,j,k} + q_{i+1/2,j,k} \\ & + q_{i,j,k-1/2} + q_{i,j,k+1/2} + QS_{i,j,k} = SS_{i,j,k} \frac{\Delta h_{i,j,k}}{\Delta t} \Delta r_j \Delta c_i \Delta v_k \end{aligned} \quad (22)$$

where

$\frac{\Delta h_{i,j,k}}{\Delta t}$  is a finite-difference approximation for head change with respect to time ( $Lt^{-1}$ );

$SS_{i,j,k}$  is the specific storage of cell  $i,j,k$  ( $L^{-1}$ ); and

$\Delta r_j \Delta c_i \Delta v_k$  is the volume of cell  $i,j,k$  ( $L^3$ ).

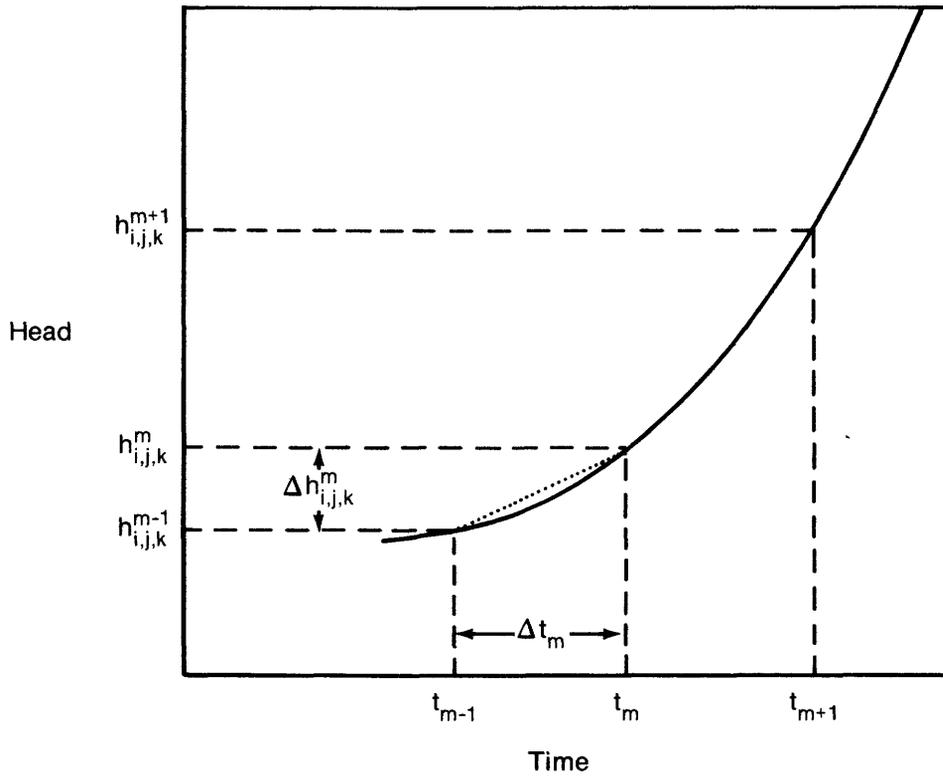
Equations 10 through 15 and 21 may be substituted into equation 22 to give the finite-difference approximation for cell  $i,j,k$  as

$$\begin{aligned} & CR_{i,j-1/2,k}(h_{i,j-1,k} - h_{i,j,k}) + CR_{i,j+1/2,k}(h_{i,j+1,k} - h_{i,j,k}) \\ & + CC_{i-1/2,j,k}(h_{i-1,j,k} - h_{i,j,k}) + CC_{i+1/2,j,k}(h_{i+1,j,k} - h_{i,j,k}) \\ & + CV_{i,j,k-1/2}(h_{i,j,k-1} - h_{i,j,k}) + CV_{i,j,k+1/2}(h_{i,j,k+1} - h_{i,j,k}) \\ & + P_{i,j,k} h_{i,j,k} + Q_{i,j,k} = SS_{i,j,k} (\Delta r_j \Delta c_i \Delta v_k) \Delta h_{i,j,k} / \Delta t. \end{aligned} \quad (23)$$

The head difference  $\Delta h_{i,j,k}$  must next be expressed in terms of specific head values which are related to the head values used to calculate flows into and out of the cell. On the hydrograph for cell  $i,j,k$  (fig. 6), two values of time,  $t_m$  and  $t_{m-1}$ , are noted on the horizontal axis; the corresponding head values,  $h_{i,j,k}^m$  and  $h_{i,j,k}^{m-1}$ , are indicated on the vertical axis; the slope of the dotted line is  $\Delta h_{i,j,k}^m / \Delta t_m$ . In the method of computation utilized here, the flow terms of equation 23 are evaluated at the more advanced time,  $t_m$ , while the hydrograph slope,  $\Delta h / \Delta t$ , is evaluated as

$$\frac{\Delta h_{i,j,k}^m}{\Delta t_m} = \frac{h_{i,j,k}^m - h_{i,j,k}^{m-1}}{t_m - t_{m-1}}. \quad (24)$$

Thus the hydrograph slope, or time derivative, is approximated using the change in head at the node over a time interval which precedes, and ends with, the time at which flow is evaluated. This is termed a backward-difference approach, in that  $\Delta h / \Delta t$  is calculated over a time interval which extends backward in time from  $t_m$ , the time at which the flow terms are evaluated. There are other ways in which  $\Delta h / \Delta t$  could be approximated; for example, we could approximate it over a time interval which begins at the time of flow evaluation and extends to some later time or over a time interval which is centered at the time of flow evaluation extending both forward and backward from it. However, there can be problems of numerical instability using these alternatives. Numerical instability means that if heads are calculated at successive times, and if for any reason errors enter the calculation at a particular time, these errors will increase at each succeeding time as the calculation progresses until finally they completely dominate the result. By contrast, the backward-difference approach is always numerically stable--



#### Explanation

- $t_m$  time at end of time step  $m$
- $h_{i,j,k}^m$  head at node  $i,j,k$  at time  $t_m$
- ..... Backward difference approximation to slope of hydrograph at time  $t_m$

Figure 6.—Hydrograph for cell  $i,j,k$ .

that is, errors introduced at any time diminish progressively at succeeding times. For this reason, the backward-difference approach is preferred even though it leads to large systems of equations which must be solved simultaneously for each time at which heads are to be computed.

Equation 23 can be rewritten in backward-difference form by specifying flow terms at  $t_m$ , the end of the time interval, and approximating the time derivative of head over the interval  $t_{m-1}$  to  $t_m$ ; that is,

$$\begin{aligned}
 & CR_{i,j-1/2,k}(h_{i,j-1,k}^m - h_{i,j,k}^m) + CR_{i,j+1/2,k}(h_{i,j+1,k}^m - h_{i,j,k}^m) \\
 & + CC_{i-1/2,j,k}(h_{i-1,j,k}^m - h_{i,j,k}^m) + CC_{i+1/2,j,k}(h_{i+1,j,k}^m - h_{i,j,k}^m) \\
 & + CV_{i,j,k-1/2}(h_{i,j,k-1}^m - h_{i,j,k}^m) + CV_{i,j,k+1/2}(h_{i,j,k+1}^m - h_{i,j,k}^m) \\
 & + P_{i,j,k}h_{i,j,k}^m + Q_{i,j,k} = SS_{i,j,k}(\Delta r_j \Delta c_i \Delta v_k) \frac{(h_{i,j,k}^m - h_{i,j,k}^{m-1})}{t_m - t_{m-1}}. \quad (25)
 \end{aligned}$$

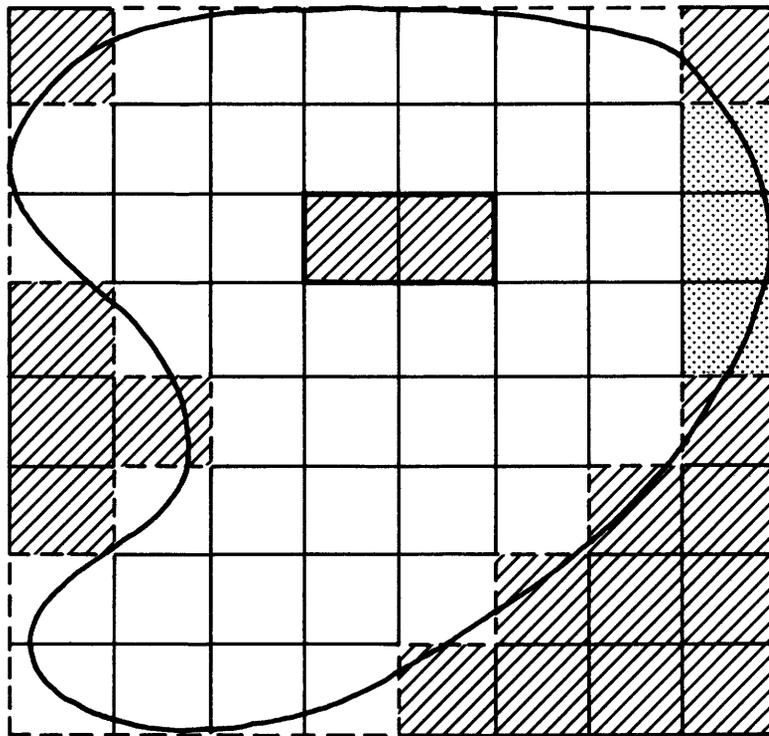
In this equation, head at the beginning of the time step  $h_{i,j,k}^m$  and all conductances and coefficients related to the node at  $i,j,k$  are known. The seven heads at time  $t_m$ , the end of the time step, are unknown; that is, they are part of the head distribution we are trying to predict. Thus equation 25 cannot be solved independently since it represents a single equation in seven unknowns. However, an equation of this type can be written for each of the "n" cells in the system; and, since there is only one unknown head for each cell, we are left with a system of "n" equations in "n" unknowns. Such a system of equations can be solved simultaneously.

In most cases, the actual number of equations will be less than the total number of model cells. The number of equations is equal to the number of "variable-head cells." Variable-head cells are those in which head

may vary with time. An equation of the form of equation 25 is required for each variable-head cell. Cells that are not variable-head cells may be either constant head or no flow. Constant-head cells are those in which head remains constant with time and, as such, do not require an equation. The equation for adjacent, variable-head cells, however, will contain nonzero conductance terms representing flow from the constant-head cell. No-flow cells are those to which there is no flow from adjacent cells. Not only is there no equation formulated for a no-flow cell but equations for adjacent cells will not contain a term representing flow from the no-flow cell.

The different cell types are used to represent various types of boundaries. In general, the types of boundaries that may be imposed in the model include constant-head, no-flow, constant-flow, and head-dependent flow. An example of the use of no-flow and constant-head cells to simulate boundary conditions is given in figure 7. Constant-flow and head-dependent flow boundaries are represented by a combination of no-flow cells and external sources.

The objective of transient simulation is to predict head patterns at successive times when given the initial-head distribution and the boundary conditions. The initial-head distribution consists of a value of  $h^1_{i,j,k}$  at each point in the mesh at time  $t_1$ , the beginning of the first of the discrete time steps into which the time axis is divided in the finite-difference process. The first step in the solution process is to calculate values of  $h^2_{i,j,k}$ --that is, heads at time  $t_2$  which mark the end of the first time step. In equation 25, therefore, the subscript  $m$  is taken as 2, while the subscript  $m-1$ , which appears in only one head term, is taken as 1. The equation therefore becomes



Explanation

- Aquifer Boundary
- - - Model Impermeable Boundary
-  Inactive Cell
-  Constant-Head Cell
-  Variable-Head Cell

Figure 7.—Discretized aquifer showing boundaries and constant-head cells.

$$\begin{aligned}
& CR_{i,j-1/2,k}(h_{i,j-1,k}^2 - h_{i,j,k}^2) + CR_{i,j+1/2,k}(h_{i,j+1,k}^2 - h_{i,j,k}^2) \\
& + CC_{i-1/2,j,k}(h_{i-1,j,k}^2 - h_{i,j,k}^2) + CC_{i+1/2,j,k}(h_{i+1,j,k}^2 - h_{i,j,k}^2) \\
& + CV_{i,j,k-1/2}(h_{i,j,k-1}^2 - h_{i,j,k}^2) + CV_{i,j,k+1/2}(h_{i,j,k+1}^2 - h_{i,j,k}^2) \\
& + P_{i,j,k}h_{i,j,k}^2 + Q_{i,j,k} \\
& = SS_{i,j,k} \frac{(\Delta r_j \Delta c_i \Delta v_k)(h_{i,j,k}^2 - h_{i,j,k}^1)}{t_2 - t_1}. \tag{26}
\end{aligned}$$

When the heads for time  $t_2$  have been obtained, the process is repeated to obtain heads at time  $t_3$ , the end of the second time step. To do this, equation 25 is reapplied, now using 2 as time subscript  $m-1$  and 3 as time subscript  $m$ . Again, a system of  $n$  equations in  $n$  unknowns is formulated where the unknowns are now the heads at  $t_3$ ; and this set of equations is solved simultaneously to obtain the head distribution at  $t_3$ . This process is continued for as many time steps as necessary to cover the time range of interest.

It is important to note that the set of finite-difference equations is reformulated at each time step; that is, at each step there is a new system of simultaneous equations to be solved. The heads at the end of the time step make up the unknowns for which this system must be solved; the heads at the beginning of the step are among the known terms in the equations. The solution process is repeated at each time step yielding a new array of heads for the end of the time step.

## Iteration

The solution at each time step might be obtained by direct algebraic methods--for example, by some procedure of eliminating variables. This would yield an "exact" solution to the set of finite-difference equations in the sense that the only factor limiting the accuracy of the results would be the number of places to which the arithmetic was carried in calculating the head values. While such direct algebraic methods are sometimes used to calculate head values at the end of a time step, numerical problems associated with their use tend to make them less desirable than iterative methods.

An iterative method starts with an initial trial solution. A procedure of calculation is then initiated which uses the trial solution to calculate an interim solution which more nearly satisfies the system of equations. The interim solution then becomes the new trial solution and the procedure is repeated. Each repetition is called an "iteration." The process is repeated until it "closes"; that is, until an iteration occurs in which the trial solution and the interim solution are "nearly" equal. The trial solution and interim solution are said to be "nearly" equal if, for each node, the difference between the trial-head value and the interim-head value is smaller than some arbitrarily established value, usually termed the "closure criterion." The interim solution is then regarded as a good approximation to the solution of the system of equations. Thus during a time step, arrays of interim-head values are generated in succession, each array containing one interim-head value for each node. In figure 8, these arrays are represented by three-dimensional lattice symbols with a superscript used to indicate the level of iteration. Thus  $h_{i,j,k}^{m,0}$  represents the initial trial value chosen for head at node  $i,j,k$ ; and  $h_{i,j,k}^{m,1}$  is the interim head calculated during

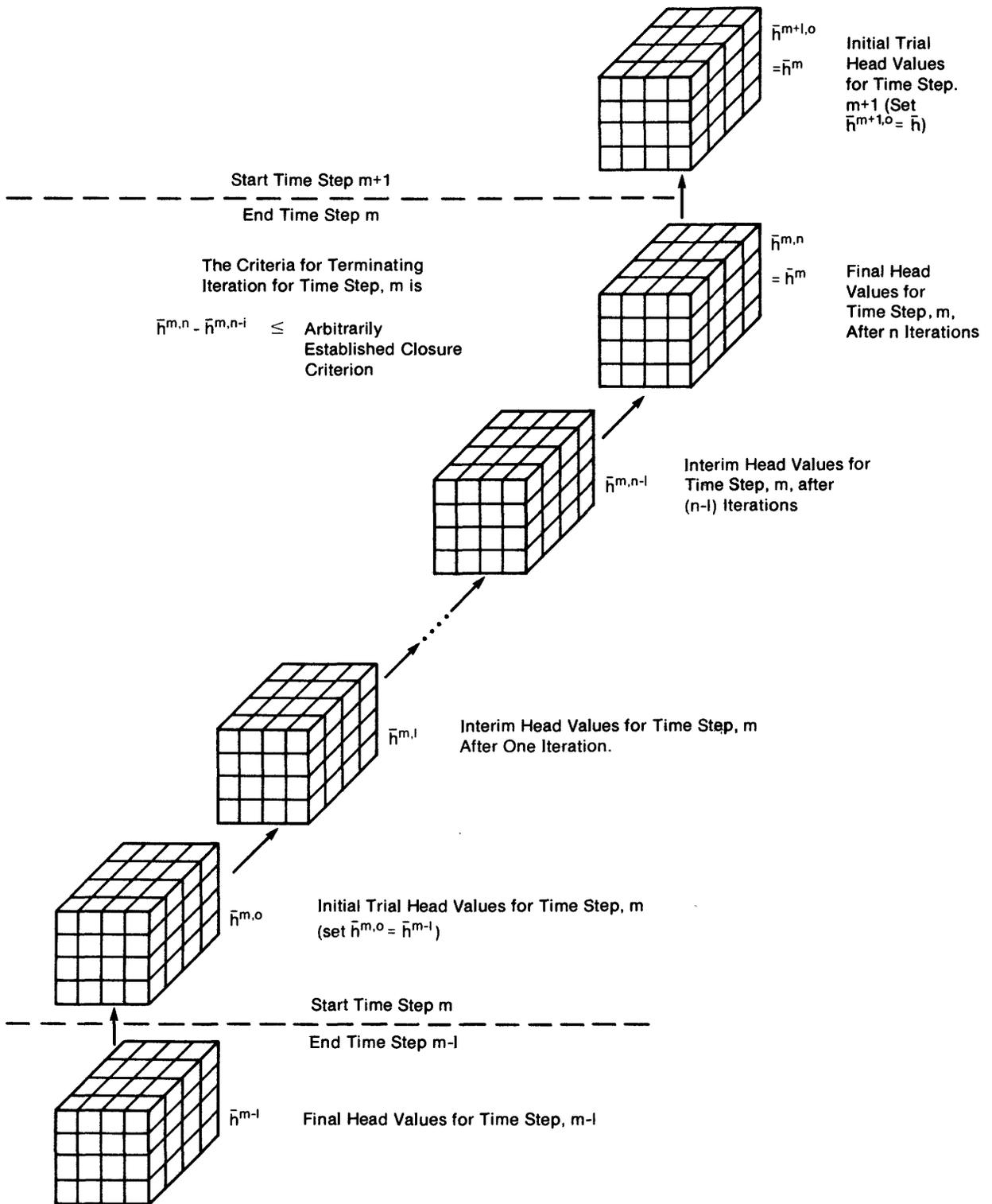


Figure 8.—Iterative calculation of a head distribution.

iteration one and the trial value used for iteration two. Similarly,  $h_{i,j,k}^{m,2}$  is the interim solution from iteration two and the trial value for iteration three.

For time  $t_m$ , the values of  $h_{i,j,k}^{m-1}$  (the final head obtained for the end of the preceding time step) are used in the storage term. These head terms for the preceding time step appear in the equation as constants; thus they retain the same values from one iteration to the next and are not modified in the iterative process. When the process is complete for time  $t_m$ , calculations for the time  $t_{m+1}$  are initiated. The final head values computed for time  $t_m$  then become the fixed-head values in the storage term used to calculate heads at time  $t_{m+1}$ .

As the preceding discussion indicates, the iterative procedure yields only an approximation to the solution of the system of finite-difference equations for each time step; the accuracy of this approximation depends upon the closure criterion which is employed. However, it is important to note that even if exact solutions to the set of finite-difference equations were obtained at each step, these exact solutions would themselves be only an approximation to the solution of the differential equation of flow (eq. 1). The discrepancy between the head,  $h_{i,j,k}^m$ , given by the solution to the system of difference equations for a given node and time, and the head  $h(x_i, y_j, z_k, t_m)$  which would be given by the formal solution of the differential equation for the corresponding point and time, is termed the truncation error. In general, it becomes greater as the mesh spacing and time-step length are increased. Finally, it must be recognized that even if a formal solution of the differential equation could be obtained, it would normally be only an approximation to conditions in the field, in that hydraulic

conductivity and specific storage are seldom known with accuracy and uncertainties with regard to hydrologic boundaries are generally present.

In summary, flow can be simulated by writing the continuity equation for each cell (eq. 25), and solving the resulting system of algebraic equations for head at each node. It is convenient to rearrange equation 25 so that all terms containing heads at the end of the current time step are grouped on the left hand side of the equation and all terms that are independent of head at the end of the current time step are on the right hand side of the equation. The new equation is given by

$$\begin{aligned}
 & CV_{i,j,k-1/2} h_{i,j,k-1}^m + CC_{i-1/2,j,k} h_{i-1,j,k}^m + CR_{i,j-1/2,k} h_{i,j-1,k}^m \\
 & + (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
 & - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k}) h_{i,j,k}^m + CR_{i,j+1/2,k} h_{i,j+1,k}^m \\
 & + CC_{i+1/2,j,k} h_{i+1,j,k}^m + CV_{i,j,k+1/2} h_{i,j,k+1}^m = RHS_{i,j,k} \quad (27)
 \end{aligned}$$

where

$$\begin{aligned}
 HCOF_{i,j,k} &= P_{i,j,k} - SC1_{i,j,k} / (t_m - t_{m-1}); & (L^2 t^{-1}) \\
 RHS_{i,j,k} &= -Q_{i,j,k} - SC1_{i,j,k} h_{i,j,k}^{m-1} / (t_m - t_{m-1}); \text{ and } & (L^3 t^{-1}) \\
 SC1_{i,j,k} &= SS_{i,j,k} \Delta r_j \Delta c_i \Delta v_k. & (L^2)
 \end{aligned}$$

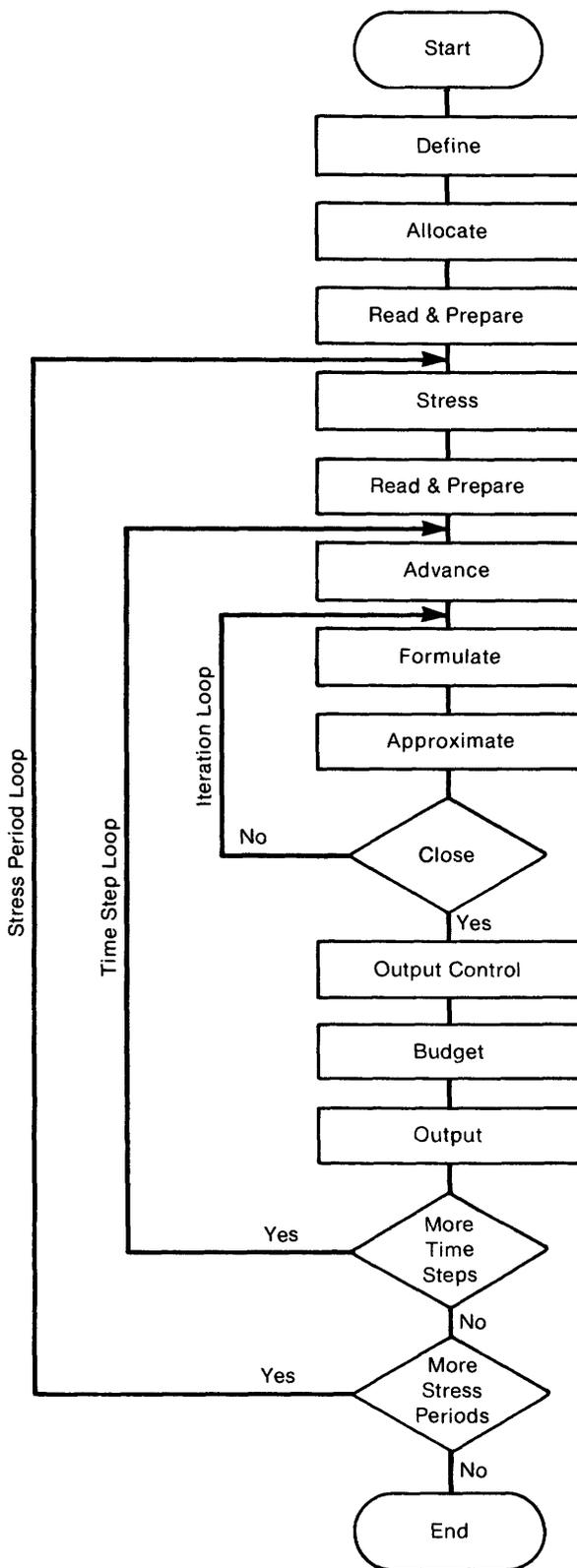
Equation 27 is the finite-difference equation that is used to develop the system of linear equations from which head is calculated and is the basis of the ground-water flow model.

## CHAPTER 3

### PROGRAM DESIGN

This chapter describes the overall design of the program. The program consists of a main program (MAIN) and a large number of highly independent subroutines called modules. This chapter will explain the functions of MAIN and explain how the modules are organized into "packages" and "procedures."

The functions which must be performed for a typical simulation are shown in figure 9. The period of simulation is divided into a series of "stress periods" within which all external stresses are constant. Each stress period, in turn, may be divided into a series of time steps. The system of finite-difference equations of the form of equation 27 is formulated and solved to produce head at each node at the end of each time step. An iterative solution method is generally used to solve for the heads for each time step. Thus within a simulation, there are three nested loops: a stress-period loop within which there is a time-step loop which, in turn, contains an iteration loop. Each rectangle in the figure is termed a "procedure." For example, prior to entering the stress loop, the program executes three procedures which pertain to a simulation as a whole. In the Define Procedure, the problem to be simulated is defined: the size of the model, the type of simulation (transient or steady-state), the number of stress periods, the hydrologic options, and the solution scheme desired are specified. In the Allocate Procedure, memory space required by the program is allocated. In the Read and Prepare Procedure, all data that are not functions of time are read. These data may include all or some of the following: boundary conditions, initial heads, transmissivity/hydraulic conductivity, specific yield/storage coefficients, elevations of layer



DEFINE — Read data specifying number of rows, columns, layers, stress periods, and major program options.

ALLOCATE — Allocate space in the computer to store data.

READ AND PREPARE — Read data which is constant throughout the simulation. Prepare the data by performing whatever calculations can be made at this stage.

STRESS — Determine the length of a stress period and calculate terms to divide stress periods into time steps.

READ AND PREPARE — Read data which changes from one stress period to the next. Prepare the data by performing whatever calculations can be made at this stage.

ADVANCE — Calculate length of time step and set heads at beginning of a new time step equal to heads calculated for the end of the previous time step.

FORMULATE — Calculate the coefficients of the finite difference equations for each cell.

APROXIMATE — Make one cut at approximating a solution to the system of finite difference equations.

OUTPUT CONTROL — Determine whether results should be written or saved on disk for this time step. Send signals to the BUDGET and OUTPUT procedures to indicate exactly what information should be put out.

BUDGET — Calculate terms for the overall volumetric budget and calculate and save cell-by-cell flow terms for each component of flow.

OUTPUT — Print and save heads, drawdown and overall volumetric budgets in accordance with signals from OUTPUT CONTROL procedure.

Figure 9.—Overall program structure.

tops and bottoms, and parameters required by the specified solution scheme. Certain preliminary calculations are also made in this procedure to prepare data for further processing.

In the Stress Procedure, the number of time steps (NSTP) in the stress period and information to calculate the length of each time step is read. In a second Read and Prepare Procedure, all data that pertain to a stress period such as pumping rates and areal recharge are read and processed. The time-step loop is then entered (fig. 9). In the Advance Procedure, the length of the time step is calculated and the heads for the start of the time step are initialized. The iteration loop contains the Formulate Procedure which determines the conductances and coefficients for each node as required by equation 27 and the Approximate Procedure which approximates a solution to the system of linear equations for head. Iteration proceeds until closure is achieved or until a specified maximum number of allowable iterations is reached. At the end of the iteration loop, the Output Control Procedure determines the disposition of the computed heads, budget terms, and cell-by-cell flow terms. In the Budget Procedure, budget entries are calculated and cell-by-cell flow terms are printed or recorded. In the Output Procedure, heads, drawdown, and the volumetric budget are printed or recorded.

Each of the modules into which the program is divided is contained within a single procedure. All modules that allocate space will fall into the Allocate Procedure; all modules that formulate the equations fall into the Formulate Procedure. Thus all of the modules can be grouped by the procedure in which they are contained.

Figure 9 is a flow chart of the overall structure of the program. It is also the flow chart for the main program. The work within the rectangles is performed by individual modules which are called by MAIN. Thus MAIN is an organized collection of FORTRAN CALL statements which invoke modules to read data, perform calculations, and print results. MAIN does not do work; it merely calls modules which do the work. The modules called directly by MAIN are called "primary" modules. Another class of modules, called "secondary" modules, are called by primary modules or other secondary modules.

Modules can be grouped by "procedure." They can also be grouped by "package." In general, a package consists of all modules associated with a particular hydrologic feature, a solution method, or the overall control of the simulation. For example, each of the modules concerned with the simulation of rivers are members of a single package--the River Package. Similarly, there are packages to simulate the effect of wells, areal recharge, drains, evapotranspiration, and general-head boundaries (table 1). All modules related to internal flow between model cells and flow into storage for a block-centered formulation are members of the Block-Centered Flow Package. The packages related to internal and external flow are termed "Flow-Component" Packages. Flow-Component Packages add terms to the finite-difference equations. Another set of packages termed the "Solver" Packages include modules needed to implement a particular solution algorithm. These packages include the Strongly Implicit Procedure (SIP) and Slice-Successive Overrelaxation (SSOR). The Solver and Flow-Component Packages are, in effect, the options available to the users of the model; that is, the user specifies which of the Flow-Component Packages are required for a simulation and which Solver Package is desired. Another package, the Basic Package (table 1), is used in any simulation irrespective of the options selected. It includes those modules

Table 1.--List of packages.

<u>Package Name</u>	<u>Abbreviation</u>	<u>Package Description</u>
Basic	BAS	Handles those tasks that are part of the model as a whole. Among those tasks are: specification of boundaries, determination of time step length, establishment of initial conditions, and printing of results.
Block-Centered Flow	BCF	Calculates terms of finite-difference equations which represent flow within the porous medium; specifically, flow from cell to cell and flow into storage.
Well	WEL	Adds terms representing flow to wells to the finite-difference equations.
Recharge	RCH	Adds terms representing areally distributed recharge to the finite-difference equations.
River	RIV	Adds terms representing flow to or from rivers to the finite-difference equations.
Drain	DRN	Adds terms representing flow to drains to the finite-difference equations.
Evapotranspiration	EVT	Adds terms representing ET to the finite-difference equations.
General-Head Boundaries	GHB	Adds terms representing general-head boundaries to the finite-difference equations.
Strongly Implicit Procedure	SIP	Iteratively solves the system of finite-difference equations using the Strongly Implicit Procedure.
Slice-Successive Overrelaxation	SOR	Iteratively solves the system of finite-difference equations using slice-successive overrelaxation.

which initialize and organize a simulation. For example, it handles initial conditions, boundary conditions, and discretization of the aquifer into cells.

In figure 10, the primary modules (subroutines called from the main program) are arranged in a matrix format to illustrate the classification by package and by procedure. The horizontal rows in the matrix correspond to procedures, while the vertical columns correspond to packages. An "X" is entered in each block of the matrix for which a module exists; absence of an "X" means that a module representing that particular package and procedure is not required. Entries marked with a subscript "S" indicate primary modules which utilize submodules in accomplishing their function. Submodules are secondary modules contained in a particular package. Entries marked with the subscript "U" indicate modules which utilize utility modules. Utility modules are secondary modules available to many packages.

The primary modules are named according to a convention which indicates both the package and the procedure to which they belong. The first three characters designate the package, the fourth is a package version number, and the last two, the procedure. For example, in figure 10, a module is indicated that is part of the Well Package and Allocate Procedure. This module is designated as WEL1AL and is a primary module that belongs to the Well Package, as indicated by the first three letters of its designation, and to the Allocate Procedure, as designated by the last two letters. It is one of the modules that deals with the simulation of specified withdrawal or input, as through wells. Its particular function is to allocate space in computer memory used to store well data. The number one appearing in the fourth place of the six-character module designation is a package version number. If the package is modified to effect improvements, a

Procedures	Flow Component Packages								Solver Packages	
	Stress Packages									
	B A S	B C F	W E L	R C H	R I V	D R N	E V T	G H B	S I P	S O R
Define (DF)	X									
Allocate (AL)	X	X	X	X	X	X	X	X	X	X
Read & Prepare (RP)	X <sub>U</sub>	X <sub>US</sub>							X	X
Stress (ST)	X									
Read & Prepare (RP)			X	X <sub>U</sub>	X	X	X <sub>U</sub>	X		
Advance (AD)	X									
Formulate (FM)	X	X <sub>S</sub>	X	X	X	X	X	X		
Approximate (AP)									X <sub>S</sub>	X <sub>S</sub>
Output Control (OC)	X									
Budget (BD)		X <sub>US</sub>	X <sub>U</sub>							
Output (OT)	X <sub>U</sub>									

Figure 10.—Organization of modules by procedures and packages.

different integer would be used in this place to distinguish the modified package from the original or from other modified versions.

Figure 11 shows the names of the primary modules arranged in the same matrix format that was used in figure 10. As in figure 10, a subscript "S" indicates that submodules are utilized and "U" indicates that utility modules are utilized.

Submodules are designated by a six-character name in which the first character is always the letter "S." This is followed by three characters designating the package name, a numeral indicating the package version number, and a one-character mnemonic to distinguish the module from other submodules of the same package; for example, the secondary module "SBCF1C" is a submodule in version one of the Block-Centered Flow Package. Utility modules are designated by the letter "U" followed by a five-character mnemonic. For example, the secondary module "U2DREL" is a utility module which reads two-dimensional real arrays.

In summary, the modules are organized so that all primary modules that perform a similar program function are grouped together in a single procedure (fig. 12). The modules are also organized so that those that deal with a particular hydrologic feature or solution method are grouped in a single package. If an entirely new package is desired, the modules can be developed and placed in the appropriate procedures without the necessity of altering existing packages.

Packages are completely independent of each other. They can be added or removed without affecting other packages. There must, however, be a

Packages

BAS	BCF	WEL	RCH	RIV	DRN	EVT	GHB	SIP	SOR
BAS1DF									
BAS1AL	BCF1AL	WEL1AL	RCH1AL	RIV1AL	DRN1AL	EVT1AL	GHB1AL	SIP1AL	SOR1AL
BAS1RP <sub>U</sub>	BCF1RP <sub>US</sub>							SIP1RP	SOR1RP
BAS1ST									
		WEL1RP	RCH1RP <sub>U</sub>	RIV1RP	DRN1RP	EVT1RP <sub>U</sub>	GHB1RP		
BAS1AD									
BAS1FM	BCF1FM <sub>S</sub>	WEL1FM	RCH1FM	RIV1FM	DRN1FM	EVT1FM	GHB1FM		
								SIP1AP <sub>S</sub>	SOR1AP <sub>S</sub>
BAS1OC									
	BCF1BD <sub>US</sub>	WEL1BD <sub>U</sub>	RCH1BD <sub>U</sub>	RIV1BD <sub>U</sub>	DRN1BD <sub>U</sub>	EVT1BD <sub>U</sub>	GHB1BD <sub>U</sub>		
BAS1OT <sub>U</sub>									

Define (DF)

Allocate (AL)

Read & Prepare (RP)

Stress (ST)

Read & Prepare (RP)

Advance (AD)

Formulate (FM)

Approximate (AP)

Output Control (OC)

Budget (BD)

Output (OT)

P R O C E D U R E S

Figure 11.—Primary modules organized by procedure and package.

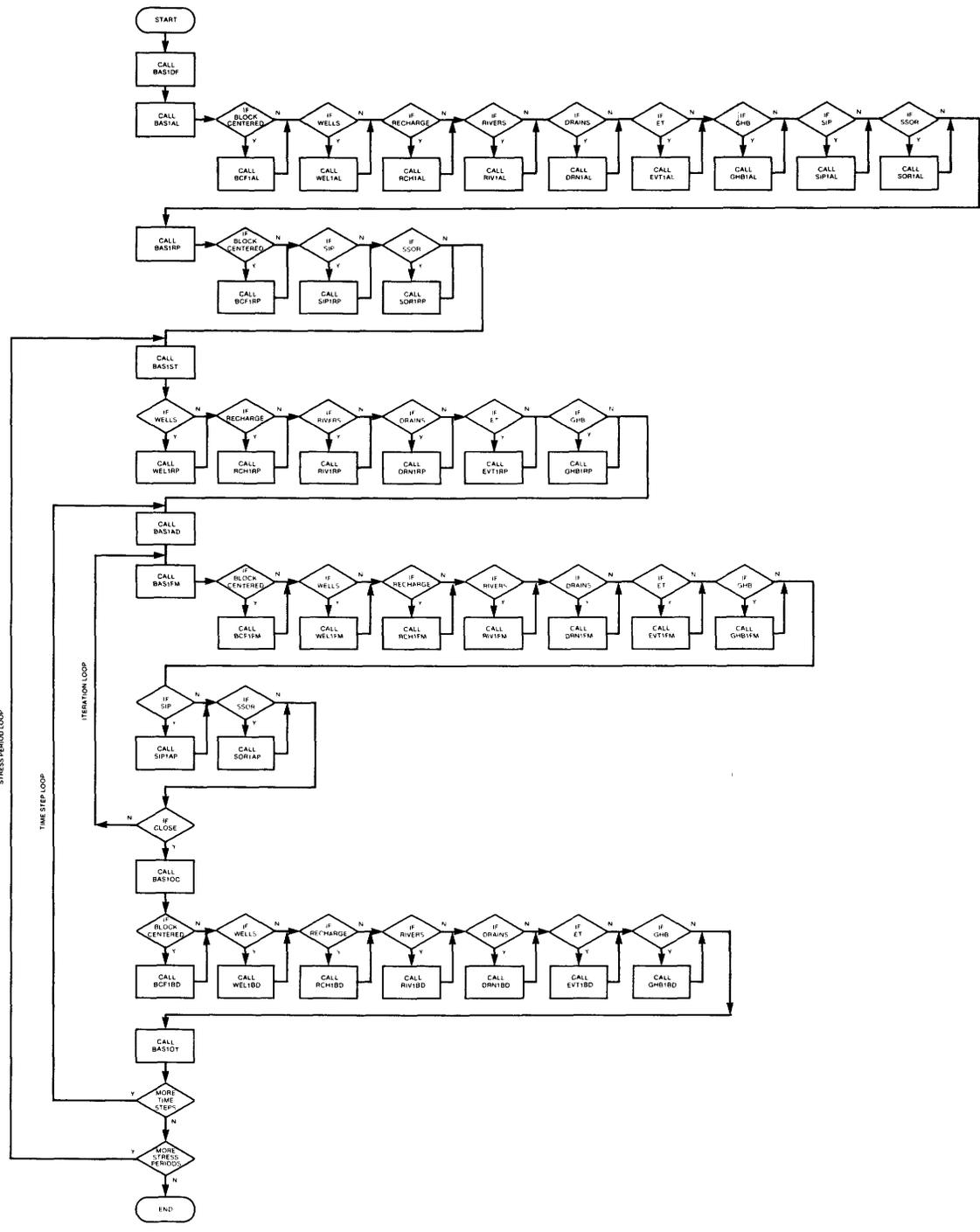


Figure 12.—Overall program structure showing all primary modules.

Basic Package, a package which calculates flow within the aquifer (Block-Centered Flow or replacement), and a Solver Package.

The organization of the program documentation parallels the package form of organization in that a separate chapter is devoted to a detailed description of each package. The remainder of this chapter describes the main program and specific topics common to all packages including boundary conditions, computer space allocation, and input/output structure.

### Boundaries

There are two types of boundaries that are integral to the model: an exterior no-flow boundary at the edges of the model grid and internal boundaries consisting of no-flow and constant-head cells. Other boundary conditions such as specified flux can be simulated as a combination of no-flow boundaries and external stresses. During formulation of equations for the first and last rows and columns of each layer, the conductance across the exterior faces are automatically set to zero. Thus it is not necessary to place no-flow boundaries at the exterior nodes of the grid. Internal no-flow and constant-head boundaries are entered by the user in the form of a code for each cell in the grid. The codes, which are stored in an array called "IBOUND," divide the cells into three disjoint sets:

IBOUND < 0-----Constant-head cell

IBOUND = 0-----Inactive cell

IBOUND > 0-----Variable-head cell

Variable-head cells are those in which the head can be expected to vary with time; a finite-difference equation is formulated for each one.

Constant-head cells are those in which head is constant throughout the simulation. Finite-difference equations are not formulated for constant-head cells. However, flow to or from constant-head cells is represented by a term in the equation of each adjoining variable-head cell. Inactive cells are those cells in which there is no flow. They are not represented in any finite-difference equation.

The IBOUND codes are initially specified by the user. If necessary, the codes are adjusted so that they are consistent with other data specified by the user and with intermediate results. For example, cells which are specified as active but are given transmissivity and vertical leakance equal to zero are changed to inactive cells.

#### Space Allocation

Space in the central memory of the computer used by data arrays and lists is allocated at execution time in a one-dimensional array called the "X" array. The Allocate Procedure contains a module for each package of the model which allocates space needed by that package. The total number of words needed in the X array depends on the type and number of packages required in a simulation and generally will range from 10 to 20 times the number of cells in the grid.

#### Input Structure

The input structure of the program is designed to permit input to be gathered, as it is needed, from many different stored files. It is based on an element of the FORTRAN language called the unit number. The unit number symbolically identifies the location of the file to be read or written.

In general, the user must provide a connection between a unit number and the name of a file by use of job control statements.

For input purposes, the program is divided into the Basic Package and several "major options." The major options generally correspond to individual packages. For example, the River Package is a major option; so is the Block-Centered Flow Package. "Output Control," which controls output from the model, is a major option even though it is part of the Basic Package.

One of the first steps in organizing input data is to specify which of the major options available are to be used. The options are specified in the "IUNIT" array (fig. 13) which is read in the Define Procedure by the Basic Package. An option is invoked by assigning a unit number to the corresponding element of the IUNIT array. If an option is not desired, the value of the element is set to zero. Thus the IUNIT array serves as a flag to indicate whether an option is active and also serves to specify the unit number containing input data required by the option. For example, if the Drain Package is used, the third element of the IUNIT array (fig. 14) is set to a nonzero unit number. In the main program, the value of IUNIT (3) is tested in several of the program procedures. If it is zero, the Drain module associated with the procedure is not called. If IUNIT (3) is greater than zero, the subroutine is called and input data is read from the file associated with the unit number.

Since the Basic Package is used for every simulation, input data of the Basic Package, are always required. Basic Package data (fig. 14) are read from unit number 1 as specified in the main program. If necessary, the unit number for BAS input can be changed to meet the requirements of a particular computer.

### Assignment of Major Options to Elements in the IUNIT Array

IUNIT	1	2	3	4	5	6	7	8	9	10	11	12
Element Number	1	2	3	4	5	6	7	8	9	10	11	12
	<i>Block Centered Flow (BCF)</i>	<i>Wells (WEL)</i>	<i>Drains (DRN)</i>	<i>Rivers (RIV)</i>	<i>Evapotranspiration (EVT)</i>	<i>Reserved for Transient Leakage</i>	<i>General Head Boundary (GHB)</i>	<i>Recharge (RCH)</i>	<i>Strongly Implicit Procedure (SIP)</i>	<i>Unused</i>	<i>Slice Successive Overrelaxation (SOR)</i>	<i>Output Control</i>

### Sample IUNIT Input Record

IUNIT	13	41	0	0	81	0	0	0	26	0	0	17
Element Number	1	2	3	4	5	6	7	8	9	10	11	12

- |                   |                     |
|-------------------|---------------------|
| 1 BCF             | Input Is on Unit 13 |
| 2 WEL             | Input Is on Unit 41 |
| 3 DRN             | Is Inactive         |
| 4 RIV             | Is Inactive         |
| 5 EVT             | Input Is on Unit 81 |
| 7 GHB             | Is Inactive         |
| 8 RCH             | Is Inactive         |
| 9 SIP             | Input Is on Unit 26 |
| 11 SOR            | Is Inactive         |
| 12 Output Control | Input Is on Unit 17 |

Figure 13.—Specification of major options using the IUNIT array.

- Options Are Specified in IUNIT. The Locations in IUNIT Are Assigned as Follows:
- 1 BCF package
  - 2 WEL package
  - 3 DRN package
  - 4 RIV package
  - 5 EVT package
  - 7 GHB package
  - 8 RCH package
  - 9 SIP package
  - 11 SOR package
  - 12 Output control

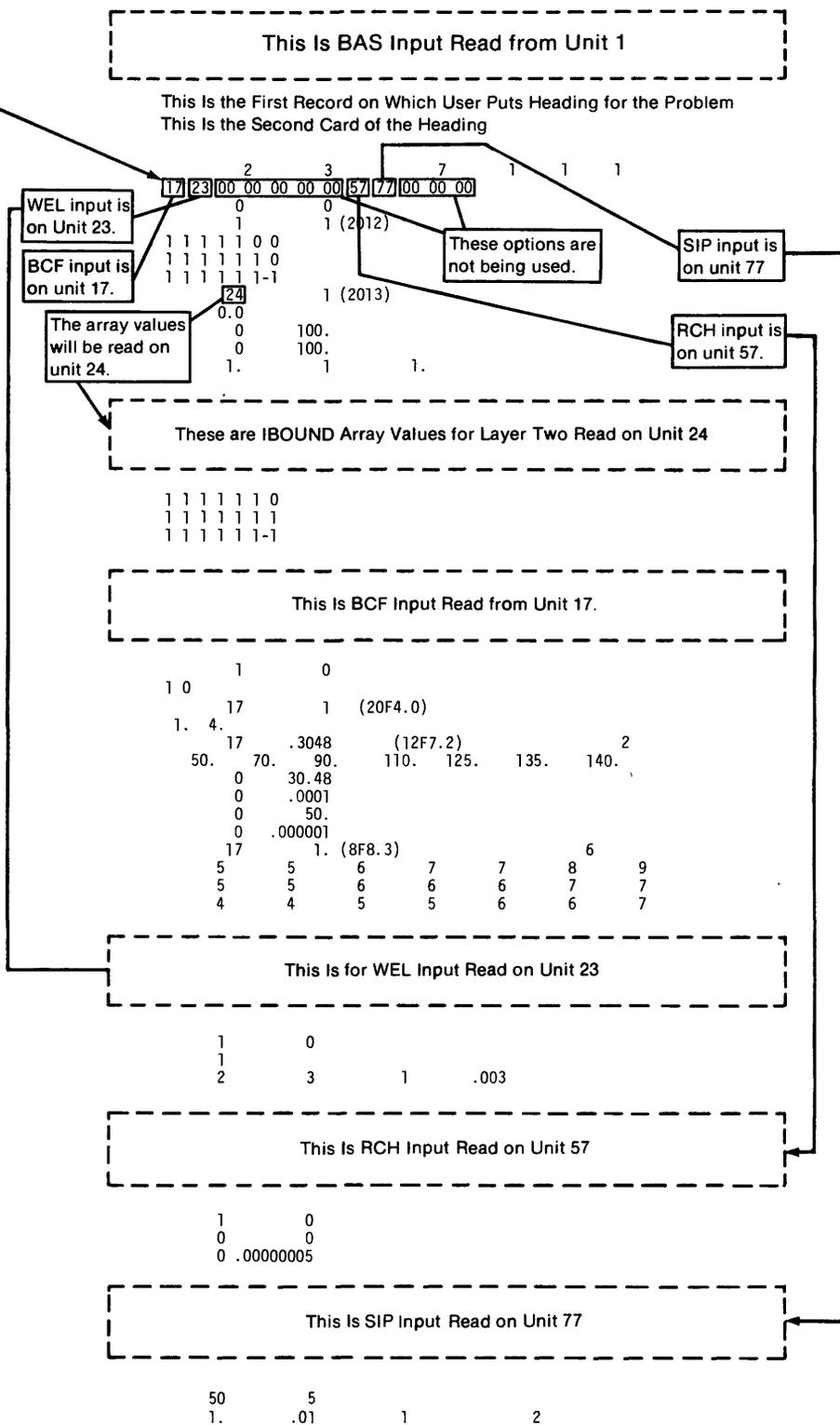


Figure 14.—Sample job showing role of the IUNIT array.

In figure 13, the Block-Centered Flow (BCF) Package is designated as being one of the available options (IUNIT(1)). As discussed in chapter 2, an alternative way of discretizing an aquifer system is the point-centered method. At present, only the BCF Package is available so that data read by this package should be considered as being required rather than an option.

Most of the data submitted by the user will consist of one-dimensional and two-dimensional arrays. Those arrays are submitted as an "array control record" plus, optionally, a series of records containing the array elements. The array control record is read from the unit number specified for the major option which calls for the array. If all the elements of an array have the same value, the value is specified on the control record and it is not necessary to read the associated array. If the elements of the array vary, records containing the array values are read from the unit specified on the control record in a format which is also specified in the control record. The unit number may be the same as that from which the control record is read or it may be different. Consequently, there is a great deal of flexibility with regard to organization of the input data required for a simulation.

Consistent length and time units must be used for all model data. The user may choose one length unit and one time unit to be used to specify all input data. This gives a certain amount of freedom to the user, but care must be exercised to avoid any mixing of units. There is no way for the program to detect the use of inconsistent units. For example, if transmissivity is entered in units of  $\text{ft}^2/\text{day}$  and pumpage as  $\text{m}^3/\text{s}$ , the program will run, but the results will be meaningless.

## Output Structure

The output structure is designed to control the amount, type, and frequency of information to be printed or written on disk. It controls the printing of head and drawdown by layer and time step, and the printing of the overall volumetric budget. It also controls disk output of head, drawdown, and cell-by-cell flow terms for use by custom-designed printing and plotting programs.

Output Control, which is a major option contained within the Basic Package, receives instructions from the user to control the amount and frequency of output. Input submitted by the user to control output is read from the unit number specified by the user for the twelfth element of the IUNIT array (IUNIT 12) at each time step. If the unit number specified by the user is equal to zero, output control information is not submitted and a default is invoked. The default output consists of head values and budget printed at the end of each stress period.

Every simulation generates some printer output. All printer output goes to unit number 6 as specified in the main program. This unit number can be changed to meet the requirements of a particular computer.

## The Main Program

The main program serves two major purposes: (1) it controls the order in which the primary modules are executed, and (2) it serves as a switching system for information. It does so with CALL statements which specify, by name, a module to be executed and lists the names of data fields (subroutine arguments) which are accessible by both the main program and the module.

The arrangement of CALL statements in the program reflects the order of procedures shown in the system flow chart (fig. 9). Within a procedure, the calls to specific modules can be in any order with one exception: if a procedure has a CALL to a module in the Basic Package, that CALL must precede all other CALLS in that procedure. Comment numbers in the listing of the main program correspond to numbers in the following list. The main program calls modules to perform tasks in the following order.

1. Set the length of the "X" array (LENX) in which all data arrays and lists are stored. Note: LENX should be set equal to the dimension of the X array prior to compilation.
2. Assign the input for the Basic Package to unit 1; assign printed output to unit 6.
3. Define the problem in terms of number of rows, columns, layers, stress periods, and major options to be used.
4. Allocate space in the X array for individual data arrays and lists.
5. If the X array is not big enough for the problem, STOP. (Redimension X and redefine LENX.)
6. Read and prepare information which is constant throughout the simulation.
7. For each stress period:
  - (a) Read stress-period timing information.
  - (b) Read and prepare information that changes each stress period.
  - (c) For each time step:

- (1) Calculate the current time-step length and move "new" heads from the preceding time step to the array containing "old" heads of the current time step.
- (2) Iteratively formulate and solve the system of equations:
  - a. Formulate the finite-difference equations.
  - b. Calculate an approximate solution to the system of equations.
  - c. If convergence criterion has been met, stop iterating.
- (3) Determine the type and amount of output needed for this time step.
- (4) Calculate overall budget terms and, if specified, calculate and print or record cell-by-cell flow terms.
- (5) Print and/or record heads and/or drawdown. Print the overall volumetric budget and timing summary.
- (6) If iteration fails to meet convergence criterion, STOP.

8. END PROGRAM.

```

C *****
C MAIN CODE FOR MODULAR MODEL -- 6/1/83
C BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C-----VERSION 1116 28DEC1983 MAIN1
C *****
C
C SPECIFICATIONS:
C -----
C COMMON X(30000)
C DIMENSION HEADNG(32),VBNM(4,20),VBVL(4,20),IUNIT(24)
C DOUBLE PRECISION DUMMY
C EQUIVALENCE (DUMMY,X(1))
C -----
C
C1-----SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
C LENX=30000
C
C2-----ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
C INBAS=1
C IOUT=6
C
C3-----DEFINE PROBLEM ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
C CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
C 1 NODES,INBAS,IOUT,IUNIT)
C
C4-----ALLOCATE SPACE IN "X" ARRAY.
C CALL BASIAL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
C 1 LCHCOF,LCRHS,LCDELRL,LCDELC,LCSTRT,LCBUFF,LCIOFL,
C 2 INBAS,ISTR,NCOL,NROW,NLAY,IOUT)
C IF(IUNIT(1).GT.0) CALL BCFIAL(ISUM,LENX,LCSC1,LCHY,
C 1 LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
C 2 NCOL,NROW,NLAY,IOUT,IBCFB)
C IF(IUNIT(2).GT.0) CALL WELIAL(ISUM,LENX,LCWELL,MXWELL,NWEL,
C 1 IUNIT(2),IOUT,IWELCB)
C IF(IUNIT(3).GT.0) CALL DRNIAL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
C 1 IUNIT(3),IOUT,IDRNCB)
C IF(IUNIT(8).GT.0) CALL RCHIAL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
C 1 NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
C IF(IUNIT(5).GT.0) CALL EVTIAL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
C 1 LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
C IF(IUNIT(4).GT.0) CALL RIVIAL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
C 1 IUNIT(4),IOUT,IRIVCB)
C IF(IUNIT(7).GT.0) CALL GHBIAL(ISUM,LENX,LCBND,NBOUND,MXBND,
C 1 IUNIT(7),IOUT,IGHBCB)
C IF(IUNIT(9).GT.0) CALL SIPIAL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,
C 1 LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
C 2 IUNIT(9),IOUT)
C IF(IUNIT(11).GT.0) CALL SORIAL(ISUM,LENX,LCA,LGRES,LCHDCG,LCLRCH,
C 1 LCIEQP,MXITER,NCOL,NROW,NLAY,NSLICE,MBW,IUNIT(11),IOUT)
C
C5-----IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
C IF(ISUM-1.GT.LENX) STOP
C
C6-----READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
C CALL BASIRP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
C 1 ISTR,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
C 2 IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
C IF(IUNIT(1).GT.0) CALL BCFIRP(X(LCIBOU),X(LCHNEW),X(LCSC1),
C 1 X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELRL),
C 2 X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
C 3 IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT)
C IF(IUNIT(9).GT.0) CALL SIPIRP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
C 1 IUNIT(9),IPCALC,IPRSIP,IOUT)
C IF(IUNIT(11).GT.0) CALL SORIRP(MXITER,ACCL,HCLOSE,IUNIT(11),
C 1 IPRSOR,IOUT)
C
C7-----SIMULATE EACH STRESS PERIOD.
C DO 300 KPER=1,NPER
C
C7A-----READ STRESS PERIOD TIMING INFORMATION.
C CALL BASIST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C7B-----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
C IF(IUNIT(2).GT.0) CALL WELIRP(X(LCWELL),NWEL,MXWELL,IUNIT(2),
C 1 IOUT)
C IF(IUNIT(3).GT.0) CALL DRNIRP(X(LCDRAI),NDRAIN,MXDRN,IUNIT(3),
C 1 IOUT)
C IF(IUNIT(8).GT.0) CALL RCHIRP(NRCHOP,X(LCIRCH),X(LCRECH),

```

```

1          X(LCDELR),X(LCDELC),NROW,NCOL,NLAY,IUNIT(8),IOUT)
IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCIEVT),X(LCEVTR),
1          X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELC),NCOL,NROW,
1          NLAY,IUNIT(5),IOUT)
IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NRIVER,MXRIVR,IUNIT(4),
1          IOUT)
IF(IUNIT(7).GT.0) CALL GH1RP(X(LCBNDS),NBOUND,MXBND,IUNIT(7),
1          IOUT)
C
C7C-----SIMULATE EACH TIME STEP.
DO 200 KSTP=1,NSTP
C
C7C1----CALCULATE TIME STEP LENGTH. SET HOLD=HNEW..
CALL BASIAD(DELTA,TSMULT,TOTIM,PRTIM,X(LCHNEW),X(LCHOLD),KSTP,
1          NCOL,NROW,NLAY)
C
C7C2----ITERATIVELY FORMULATE AND SOLVE THE EQUATIONS.
DO 100 KITER=1,MXITER
C
C7C2A---FORMULATE THE FINITE DIFFERENCE EQUATIONS.
CALL BASIFM(X(LCHCOF),X(LCRHS),NCOL,NROW,NLAY,NODES)
IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
1          X(LCSC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),X(LCCV),
2          X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCSC2),
3          X(LCDELR),X(LCDELC),DELTA,ISS,KITER,KSTP,KPER,NCOL,
4          NROW,NLAY,IOUT)
IF(IUNIT(2).GT.0) CALL WEL1FM(NWEL,MXWELL,X(LCRHS),X(LCWELL),
1          X(LCIBOU),NCOL,NROW,NLAY)
IF(IUNIT(3).GT.0) CALL DRN1FM(NDRAIN,MXDRN,X(LCDRAI),X(LCHNEW),
1          X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
IF(IUNIT(8).GT.0) CALL RCH1FM(NRCHOP,X(LCIRCH),X(LCRECH),
1          X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
IF(IUNIT(5).GT.0) CALL EVT1FM(NEVTOP,X(LCIEVT),X(LCEVTR),
1          X(LCEXDP),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
1          X(LCHNEW),NCOL,NROW,NLAY)
IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCRIVR),X(LCHNEW),
1          X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
IF(IUNIT(7).GT.0) CALL GH1FM(NBOUND,MXBND,X(LCBNDS),X(LCHCOF),
1          X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
C
C7C2B---MAKE ONE CUT AT AN APPROXIMATE SOLUTION.
IF(IUNIT(9).GT.0) CALL SI1AP(X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),
1          X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCFL),X(LCGL),X(LCV),
2          X(LCW),X(LCHDCG),X(LCLRCH),NPARM,KITER,HCLOSE,ACCL,ICNVG,
3          KSTP,KPER,IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
4          IOUT)
IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
1          X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCA),X(LCRES),X(LCIEQP),
2          X(LCHDCG),X(LCLRCH),KITER,HCLOSE,ACCL,ICNVG,KSTP,KPER,IPRSOR,
3          MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C
C7C2C---IF CONVERGENCE CRITERION HAS BEEN MET STOP ITERATING.
IF(ICNVG.EQ.1) GO TO 110
100 CONTINUE
KITER=MXITER
110 CONTINUE
C
C7C3---DETERMINE WHICH OUTPUT IS NEEDED.
CALL BASIOC(NSTP,KSTP,KPER,ISTR,ICNVG,X(LCIOFL),NLAY,
1          IBUDFL,ICBCFL,IHDDFL,IUNIT(12),IOUT)
C
C7C4---CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
MSUM=1
IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
1          X(LCIBOU),X(LCHOLD),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
2          X(LCTOP),X(LCSC2),DELTA,ISS,NCOL,NROW,NLAY,KSTP,KPER,
3          IBCFCB,ICBCFL,X(LCIBUFF),IOUT)
IF(IUNIT(2).GT.0) CALL WEL1BD(NWEL,MXWELL,VBNM,VBVL,MSUM,X(LCWELL),
1          X(LCIBOU),DELTA,NCOL,NROW,NLAY,KSTP,KPER,IWELCB,ICBCFL,
1          X(LCIBUFF),IOUT)
IF(IUNIT(3).GT.0) CALL DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,
1          X(LCDRAI),DELTA,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KSTP,KPER,
2          IDRNCB,ICBCFL,X(LCIBUFF),IOUT)
IF(IUNIT(8).GT.0) CALL RCH1BD(NRCHOP,X(LCIRCH),X(LCRECH),
1          X(LCIBOU),NROW,NCOL,NLAY,DELTA,VBVL,VBNM,MSUM,KSTP,KPER,
2          IRCHCB,ICBCFL,X(LCIBUFF),IOUT)
IF(IUNIT(5).GT.0) CALL EVT1BD(NEVTOP,X(LCIEVT),X(LCEVTR),
1          X(LCEXDP),X(LCSURF),X(LCIBOU),X(LCHNEW),NCOL,NROW,NLAY,

```

```

2   DELT,VBVL,VBNM,MSUM,KSTP,KPER,IEVTGB,ICBCFL,X(LCIBOU),IOUT)
IF(IUNIT(4).GT.0) CALL RIVIBD(NRIVER,MXRIVR,X(LCRIVR),X(LCIBOU),
1   X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
2   KSTP,KPER,IRIVCB,ICBCFL,X(LCIBOU),IOUT)
IF(IUNIT(7).GT.0) CALL GHIBD(NBOUND,MBND,VBNM,VBVL,MSUM,
1   X(LCBNDS),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KSTP,KPER,
2   IGHBCB,ICBCFL,X(LCIBOU),IOUT)
C
C7C5---PRINT AND OR SAVE HEADS AND DRAWDOWNS. PRINT OVERALL BUDGET.
CALL BASIOT(X(LCHNEW),X(LCSTRT),ISTR,X(LCIBOU),X(LCIOFL),
1   MSUM,X(LCIBOU),VBNM,VBVL,KSTP,KPER,DELT,
2   PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
3   IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
C
C7C6---IF ITERATION FAILED TO CONVERGE THEN STOP.
IF(ICNVG.EQ.0) STOP
200 CONTINUE
300 CONTINUE
C
C8-----END PROGRAM
STOP
C
END

```

## CHAPTER 4

### BASIC PACKAGE

#### Conceptualization and Implementation

The Basic Package handles the administrative tasks of the model. The major tasks for which it is responsible are the discretization of space and time into cells and time steps, specification of initial and boundary conditions, specification of heads for the beginning of each time step, specification of program options to be used, calculation of the volumetric budget, and control of the output of results.

#### Model Input and Selection of Major Options

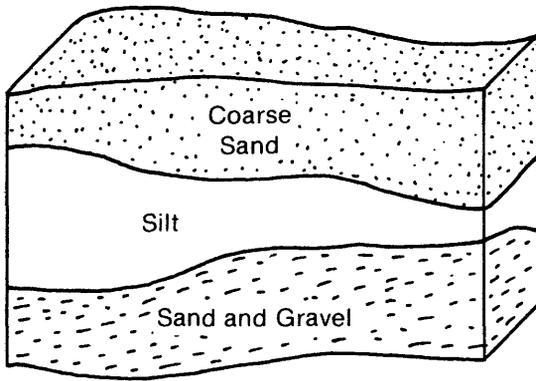
Input to the program is divided by "major option." Major options are sections of the program which the user may opt to use or not use. Major options generally correspond to packages. For example, the River Package is a major option; so is the SIP Package. The Basic Package is always used so it is not a major option. However, "Output Control," which is part of the Basic Package, is a major option. Since the Basic Package is mandatory, input to the Basic Package is always read. Input to a major option is read only if the user intends to use the option. The user selects a major option by setting the element corresponding to that option in an array named "IUNIT"--which is read by the Basic Package--equal to a positive integer. The positive integer serves two functions: (1) it indicates that the corresponding major option will be used, and (2) it is the unit number for the file containing input for that major option (fig. 13). When a new major option is added to the program, it will be assigned to an element in the IUNIT array.

## Discretization of Space

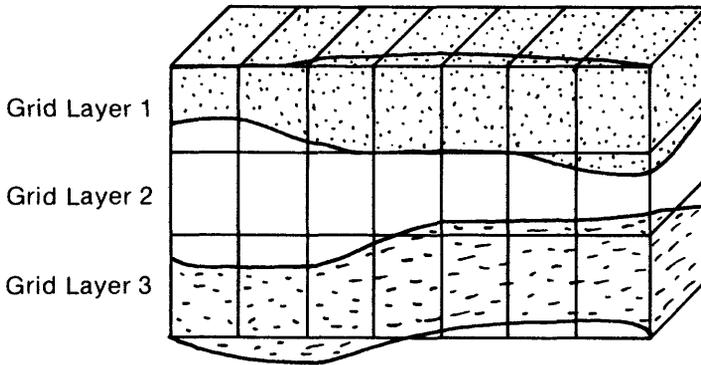
In the finite-difference method, a rectilinear grid is used to divide the region to be studied into rows, columns, and layers, forming cells with rectangular faces. The properties of the cells, which are assumed to be homogeneous, are used to formulate the coefficients of the finite-difference equations. Generally, the grid is superimposed on a flow system contained in a sequence of stratigraphic units which are not quite horizontal (fig. 15). Thus some cells may represent two very different rock types, making specification of physical properties difficult. It is convenient, therefore, to deform the grid so that grid layers follow the contours of the stratigraphic units.

Changing from a rectilinear grid to a grid based on geologic layers, though convenient, is the source of some error. Faces of each cell are no longer rectangles but irregular surfaces. However, if the layers are very nearly horizontal, the calculated heads should be very nearly correct.

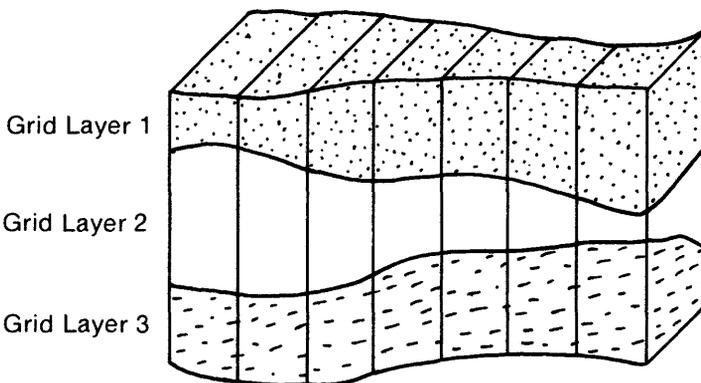
At the extreme, there are two types of geologic units which may be of interest to an investigator--high conductivity units and low conductivity units. Figure 16 shows a flow net in two high conductivity sand units separated by a low conductivity clay unit. The equipotentials in the sand units are nearly vertical; thus each of those units can be approximated accurately with just one or two layers. In the clay unit, on the other hand, the equipotentials are nearly horizontal. Therefore, many layers are needed to represent the change in head across the unit. Figure 17 shows a grid that may be needed to accurately represent head variation in the clay. In this example, the clay unit is represented by six grid layers.



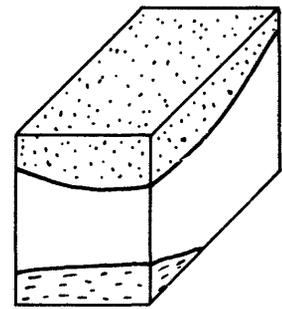
Aquifer Cross Section



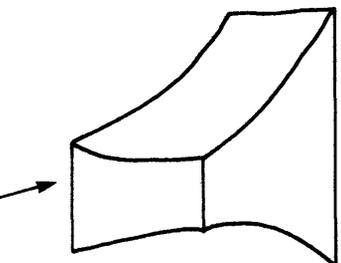
Aquifer Cross Section With Rectilinear Grid Superimposed



Aquifer Cross Section With Deformed Grid Superimposed



Cell Contains Material from Three Stratigraphic Units. All Faces Are Rectangles



Cell Contains Material from Only One Stratigraphic Unit. Faces Are Not Rectangles.

Figure 15.—Effect of using distorted grid in the vertical direction.

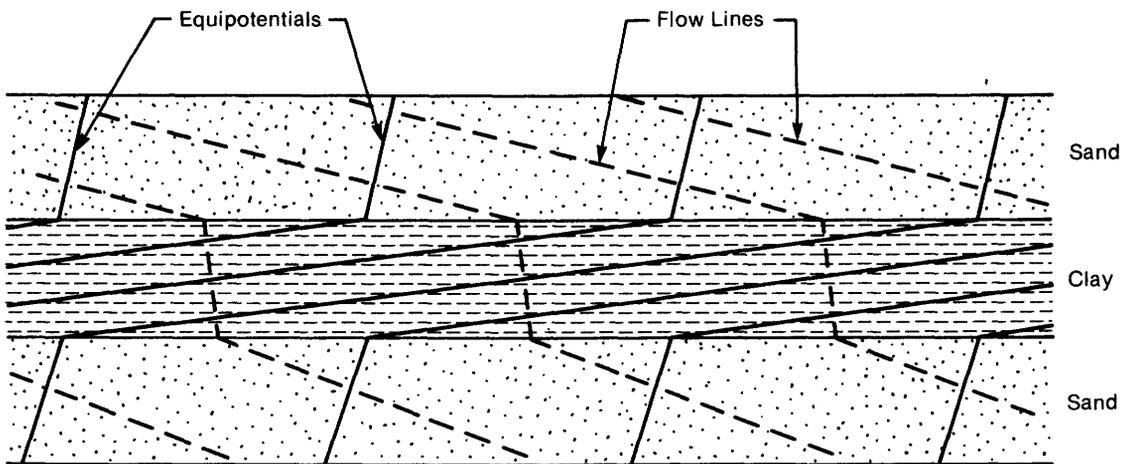


Figure 16.—Flow net in a cross section consisting of two high conductivity units separated by a low conductivity unit.

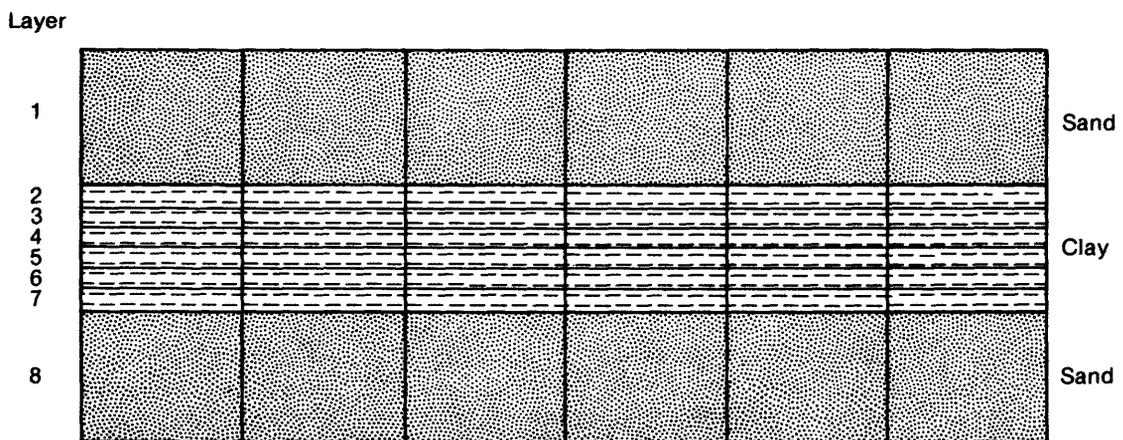


Figure 17.—A cross section in which a low conductivity unit is represented by six model layers.

The flow system illustrated in figure 17 is simulated with eight layers, one for each sand unit and six for the clay unit. However, in a similar field situation, a hydrologist would generally be more interested in heads in the sand units than those in the clay unit. Thus it may be sufficient to simulate flow within the sand units and the effect of the clay unit on transfer of water between the two sand units. Thus the upper sand unit would be layer 1 in the model; the lower sand unit would be layer 2 (fig. 18). The clay unit would not be simulated, heads in the clay would not be calculated, but properties of the clay would be used to calculate conductance between the sand layers.

In classical finite-difference theory, the vertical spacing of the grid consists of a thickness ( $\Delta v_k$ ) of each layer such that the sum of those thicknesses equals the thickness of the flow field. The two situations described above represent cases that are exceptions to the classical finite-difference method; that is, (1) grid layers, rather than being of even thickness, may be deformed to match boundaries between stratigraphic units, and (2) portions of the flow field within low conductivity units may be simulated only in as much as they affect flow between adjacent layers. When grid layers match stratigraphic units, thickness is a function of horizontal location (fig. 15). When low conductivity layers are omitted, the sum of the thicknesses of the individual simulated layers does not equal the thickness of the flow field (fig. 18).

This program handles these exceptions by incorporating layer thickness into terms representing aquifer properties. For example, in confined layers transmissivity is used rather than hydraulic conductivity and storage coefficient rather than specific storage. Consequently, vertical-grid spacing is never explicitly read by the program.

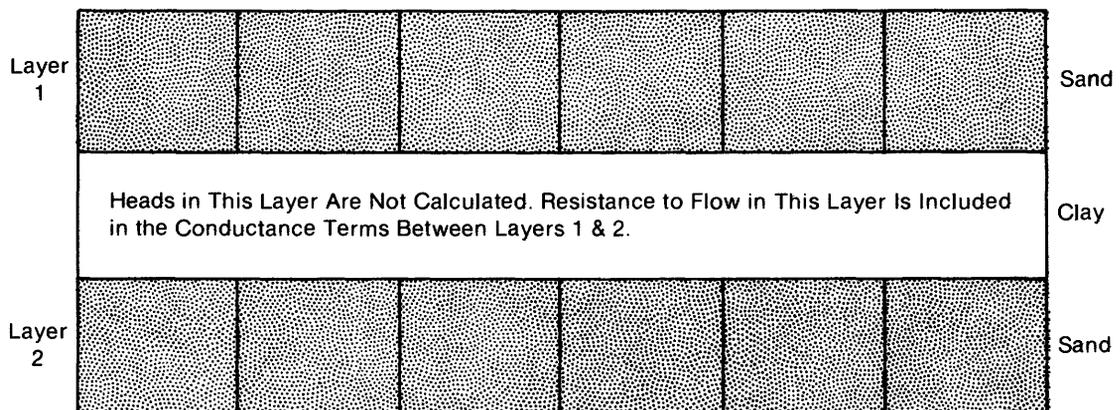


Figure 18.—A cross section in which a low conductivity unit is not represented by a model layer.

Discretization of the region to be simulated consists of specifying a number of rows, columns, layers, and the horizontal grid spacing (DELR and DELC). Grid spacing is read by the Block-Centered Flow Package. The Basic Package allocates space for horizontal grid spacing and uses the number of rows, columns, and layers to allocate space for data arrays.

### Boundaries

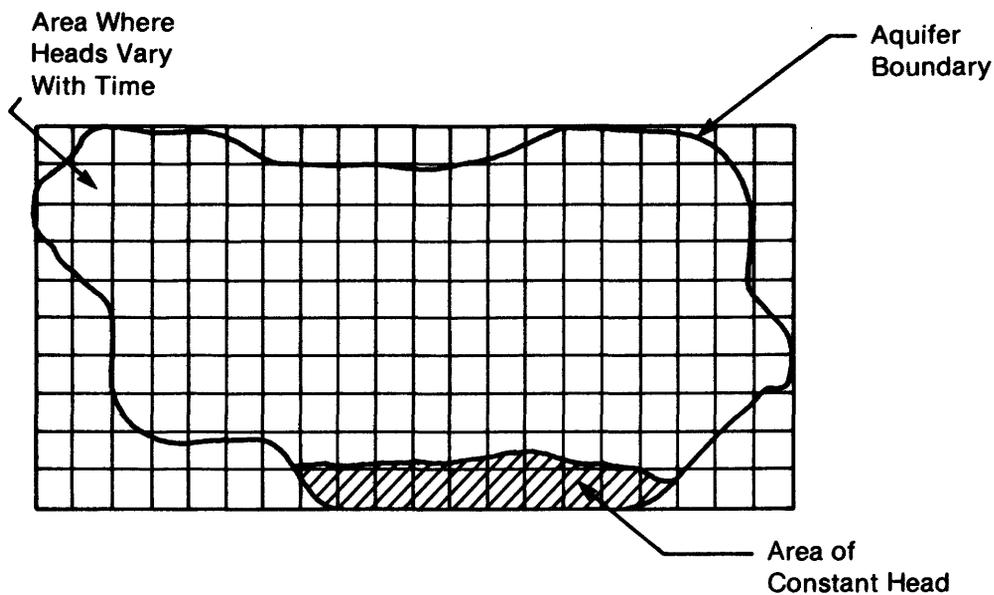
Recall that the finite-difference equation for a cell has the form

$$\begin{aligned}
 & CR_{i,j-1/2,k} (h_{i,j-1,k}^m - h_{i,j,k}^m) + CR_{i,j+1/2,k} (h_{i,j+1,k}^m - h_{i,j,k}^m) \\
 & + CC_{i-1/2,j,k} (h_{i-1,j,k}^m - h_{i,j,k}^m) + CC_{i+1/2,j,k} (h_{i+1,j,k}^m - h_{i,j,k}^m) \\
 & + CV_{i,j,k-1/2} (h_{i,j,k-1}^m - h_{i,j,k}^m) + CV_{i,j,k+1/2} (h_{i,j,k+1}^m - h_{i,j,k}^m) \\
 & + P_{i,j,k} h_{i,j,k}^m + Q_{i,j,k} = SC_{i,j,k} (h_{i,j,k}^m - h_{i,j,k}^{m-1}) / \Delta t_m. \quad (28)
 \end{aligned}$$

One finite-difference equation is written for each cell in the grid in which the head varies with time. An array, called the IBOUND array, which is specified by the user and read by the Basic Package, is used to keep track of which cells have heads which vary with time. The IBOUND array (fig. 19) contains a code for each cell which indicates whether (1) the head varies with time (variable-head cell), (2) the head is constant (constant-head cell), or (3) no flow takes place within the cell (no-flow cell). The IBOUND array can be modified by other packages if the state of a cell changes.

### Initial Conditions

Because equation 28 is in backward-difference form, a head distribution at the beginning of a time step is required to calculate the head distribution at the end of the time step (fig. 20). For each time step, the head distribution



0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0

**IBOUND Codes**  
 < 0 Constant Head  
 = 0 No Flow  
 > 0 Variable Head

Figure 19.—Example of the boundary array (IBOUND) for a single layer.

Starting heads (STRT) are the heads at the beginning of the simulation.

New Heads (HNEW) are the latest estimate of the heads at the end of the current time step. Each iteration produces a new estimate.

Old Heads (HOLD) are the heads at the beginning of the current time step. They are, therefore, equal to the heads at the end of the previous time step.

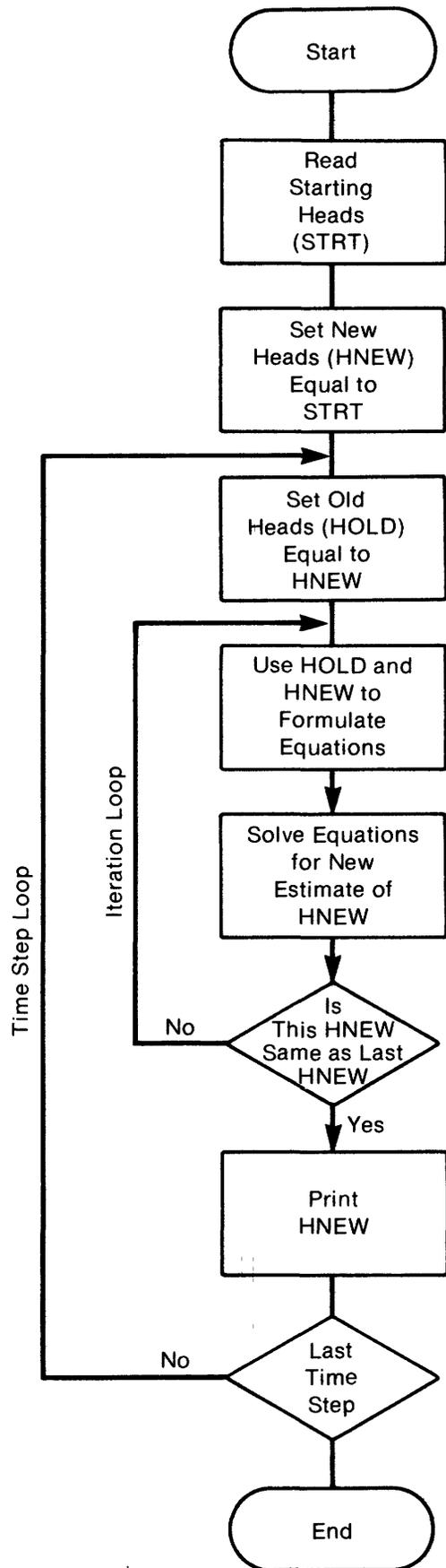


Figure 20.—Flow of head distributions during a simulation.

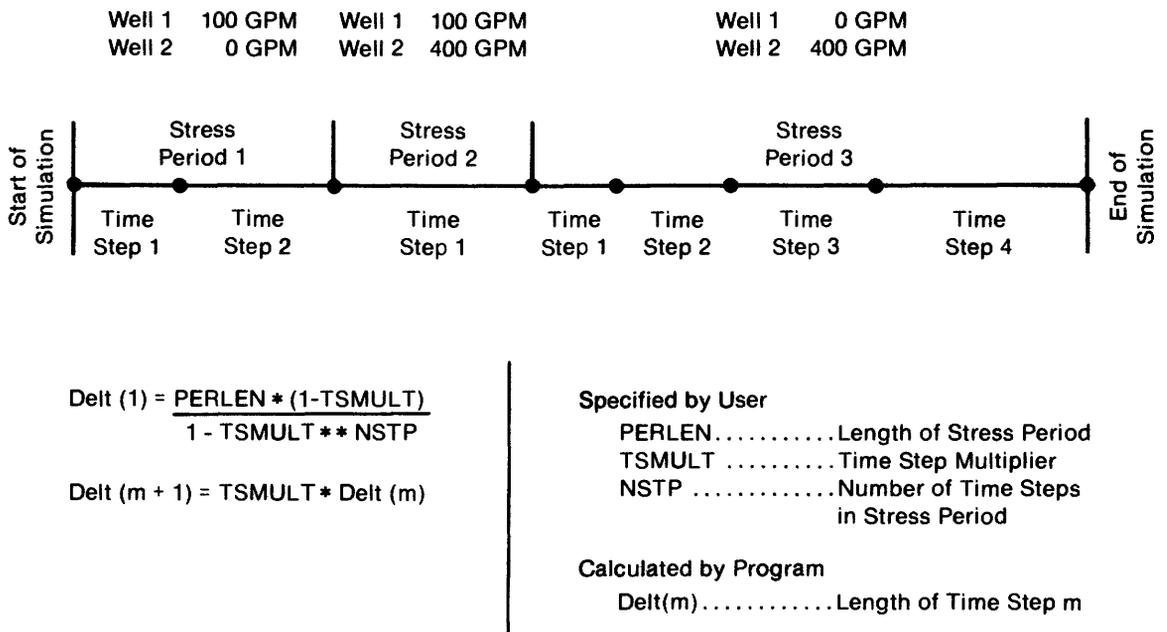
at the start of one time step is set equal to the head distribution at the end of the previous time step. That chain is started with "starting heads" specified by the user. After the first time step, starting heads are no longer used to calculate heads. They may be saved in array STRT; however, to calculate drawdown, the difference between the starting head distribution and some later head distribution.

### Discretization of Time

Simulation time is divided into stress periods--time intervals during which all external stresses are constant--which are, in turn, divided into time steps. The length of each stress period is specified explicitly by the user. Within the stress period, the time steps form a geometric series in which the parameters of the series, the number of elements, and the multiplier are specified by the user (fig. 21). The program uses those parameters along with the length of the stress period to calculate the length of each time step.

### Output

The primary output of the program is head distribution. In addition, a volumetric water budget is provided as a check on the numerical accuracy of the simulation (fig. 22). The user can also request that cell-by-cell flow terms and drawdown distributions be printed or recorded on disks. "Output Control," a major option contained in the Basic Package, is used to control the frequency and amount of data printed or saved. If Output Control is not specified, a default option is invoked--head and drawdown are printed at the end of each stress period.



**Figure 21.—Division of simulation time into stress periods and time steps.**

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

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

```

IN:
----
STORAGE = .0
CONSTANT HEAD = .0
WELLS = .0
DRAINS = .0
RECHARGE = .13608E+08
TOTAL IN = .13608E+08
OUT:
----
STORAGE = .0
CONSTANT HEAD = .43265E+07
WELLS = .64800E+07
DRAINS = .28010E+07
RECHARGE = .0
TOTAL OUT = .13607E+08
IN - OUT = 303.00
PERCENT DISCREPANCY = 0.00
    
```

```

IN:
----
STORAGE = .0
CONSTANT HEAD = .0
WELLS = .0
DRAINS = .0
RECHARGE = 157.50
TOTAL IN = 157.50
OUT:
----
STORAGE = .0
CONSTANT HEAD = 50.075
WELLS = 75.000
DRAINS = 32.419
RECHARGE = .0
TOTAL OUT = 157.49
IN - OUT = .34943E-02
PERCENT DISCREPANCY = 0.00
    
```

Figure 22.--Sample overall volumetric water budget.

The calculation of the volumetric budget consists of two parts, the calculation of the entries for the budget and the summation of the entries. The entries, which correspond to individual components of flow, are calculated in component-of-flow packages and stored in a table named VBVL. For example, total flow into rivers is calculated in the River Package; total flow to constant-head cells is calculated in the Block-Centered Flow Package. The table VBVL is passed to the Basic Package which prints and sums the budget entries.

## Basic Package Input

Input for the Basic (BAS) Package except for output control is read from unit 1 as specified in the main program. If necessary, the unit number for BAS input can be changed to meet the requirements of a particular computer. Input for the output control option is read from the unit number specified in IUNIT(12).

Information for the Basic Package must be submitted in the following order:

### FOR EACH SIMULATION

#### BAS1DF

1. Data: HEADNG(32)  
Format: 20A4
2. Data: HEADNG (continued)  
Format: 12A4
3. Data: NLAY NROW NCOL NPER ITMUNI  
Format: I10 I10 I10 I10 I10
4. Data: IUNIT(24)  
Format: 24I3  
(BCF WEL DRN RIV EVT XXX GHB RCH SIP XXX SOR OC)

#### BAS1AL

5. Data: IAPART ISTRT  
Format: I10 I10

#### BAS1RP

6. Data: IBOUND(NCOL,NROW)  
Module: U2DINT  
(One array for each layer in the grid)
7. Data: HNOFLO  
Format: F10.0
8. Data: Shead(NCOL,NROW)  
Module: U2DREL  
(One array for each layer in the grid)

### FOR EACH STRESS PERIOD

#### BAS1ST

9. Data: PERLEN NSTP TSMULT  
Format: F10.0 I10 F10.0

## Explanation of Fields Used in Input Instructions

HEADNG--is the simulation title that is printed on the printout. It may be up to 132 characters long; 80 in the first record and 52 in the second. Both records must be included even if they are blank.

NLAY--is the number of model layers.

NROW--is the number of model rows.

NCOL--is the number of model columns.

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

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

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

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

IUNIT--is a 24-element table of input units for use by all major options. Only 10 elements (1-5, 7-9, 11, and 12) are being used. Element 6 is reserved for the Transient Leakage Package. Element 10 is reserved for an additional solver. Elements 13-24 are reserved for future major options.

<u>IUNIT</u>	<u>MAJOR</u>
<u>LOCATION</u>	<u>OPTION</u>
1	Block-Centered Flow Package
2	Well Package
3	Drain Package
4	River Package
5	Evapotranspiration Package
6	Reserved for Transient Leakage Package
7	General-Head Boundary Package
8	Recharge Package
9	SIP Package
10	Reserved for additional solver
11	SSOR Package
12	Output Control Option

If  $IUNIT(n) \leq 0$ , the corresponding major option is not being used.

If  $IUNIT(n) > 0$ , the corresponding major option is being used and data for that option will be read from the unit number contained in  $IUNIT(n)$ . The unit numbers in  $IUNIT$  should be integers from 1 to 99. Although the same number may be used for all or some of the major options, it is recommended that a different number be used for each major option. Printer output is assigned to unit 6 (unless it is changed to meet computer requirements). That unit number should not be used for any other input or output. The user is also permitted to assign unit numbers for output. Those numbers should be different from those assigned to input. The Basic Package reads from unit 1 (unless it is changed to meet computer requirements). It is permissible but unwise to use that unit for other major options.

IAPART--indicates whether array BUFF is separate from array RHS.

If  $IAPART = 0$ , the arrays BUFF and RHS occupy the same space. This option conserves space. This option should be used unless some other package explicitly says otherwise.

If  $IAPART \neq 0$ , the arrays BUFF and RHS occupy different space. This option is not needed in the program as documented in this publication. It may be needed for packages yet to be written.

ISTR--indicates whether starting heads are to be saved. If they are saved, they will be stored in array STRT. They must be saved if drawdown is calculated.

If  $ISTR = 0$ , starting heads are not saved.

If  $ISTR \neq 0$ , starting heads are saved.

IBOUND--is the boundary array.

If  $IBOUND(I,J,K) < 0$ , cell I,J,K has a constant head.

If  $IBOUND(I,J,K) = 0$ , cell I,J,K is inactive.

If  $IBOUND(I,J,K) > 0$ , cell I,J,K is active.

HNOFLO--is the value of head to be assigned to all inactive cells ( $IBOUND = 0$ ) throughout the simulation. Since heads at inactive cells are unused, this does not affect model results but serves to identify inactive cells when head is printed. This value is also used as drawdown at inactive cells if the drawdown option is used. Even if the user does not anticipate having inactive cells, a value for HNOFLO must be submitted.

Shead--is head at the start of the simulation. Regardless of whether starting head is saved, these values must be input to initialize the solution.

PERLEN--is the length of a stress period. It is specified for each stress period.

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

TSMULT--is the multiplier for the length of successive time steps. The length of the first time step DELT(1) is related to PERLEN, NSTP and TSMULT by the relation

$$\text{DELT}(1) = \text{PERLEN}(1 - \text{TSMULT}) / (1 - \text{TSMULT}^{**}\text{NSTP}).$$

SAMPLE INPUT TO THE BASIC PACKAGE(EXCLUDING OUTPUT CONTROL)

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{HEADING} CONTINUED	
2	{NLAY, NROW, NCOL, NPER, ITMUNI}	
3	{IUNIT TABLE}	
4	{IAPART, ISTRT}	
5	CONTROL RECORD FOR IBOUND ARRAY LAYER 1	
6	IBOUND VALUES FOR LAYER 1	<p>SAMPLE INPUT FOR BASIC PACKAGE 3 LAYERS 15 ROWS 15 COLUMNS WITH RECHARGE, DRAINS, AND WELLS</p> <p>31 12 14 00 00 00 00 28 19 00 00 22</p> <p>0 1 (12I2)</p> <p>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</p> <p>1 1 (12I3)</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</p> <p>999.99 0 3</p> <p>10. 17. 24. 31. 38. 45. 52. 59. 66. 73. 80. 87.</p> <p>11. 19. 27. 35. 43. 51. 59. 67. 75. 83. 91. 99.</p> <p>13. 23. 33. 43. 53. 63. 73. 83. 93. 103. 113. 123.</p> <p>16. 29. 42. 55. 68. 81. 94. 107. 120. 133. 146. 159.</p> <p>20. 37. 54. 71. 88. 105. 122. 139. 156. 173. 190. 207.</p> <p>20. 37. 54. 71. 88. 105. 122. 139. 156. 173. 190. 207.</p> <p>16. 29. 42. 55. 68. 81. 94. 107. 120. 133. 146. 159.</p> <p>13. 23. 33. 43. 53. 63. 73. 83. 93. 103. 113. 123.</p> <p>11. 19. 27. 35. 43. 51. 59. 67. 75. 83. 91. 99.</p> <p>10. 17. 24. 31. 38. 45. 52. 59. 66. 73. 80. 87.</p> <p>23 1 (12F6.2)</p> <p>0 0.0</p> <p>86400 5 1.5</p> <p>172800 7 1.5</p>
6	CONTROL RECORD FOR IBOUND ARRAY LAYER 2	
7	{HNOFLO}	
8	CONTROL RECORD FOR STARTING HEAD ARRAY FOR LAYER 1	
8	STARTING HEAD VALUES FOR LAYER 1	
8	CONTROL RECORD FOR STARTING HEAD ARRAY FOR LAYER 2	
8	CONTROL RECORD FOR STARTING HEAD ARRAY FOR LAYER 3	
9	STRESS PERIOD 1---{PERLEN NSTP TSMULT}	
9	STRESS PERIOD 2---{PERLEN NSTP TSMULT}	

FIELDS IN ARRAY CONTROL RECORDS ARE---{ LOCAT, CONST, FMTIN, IPRN}

## Output Control Input

Output control is a major option separate from the rest of the Basic Package. Input to Output Control is read from the unit specified in IUNIT(12). If IUNIT(12) is zero, no output control data are read, and default output control is used. Under the default, head and total budget are printed at the end of every stress period. Additionally, if starting heads are saved (ISTRN is not 0), drawdown is printed at the end of every stress period. The default printout format for head and drawdown is 10G11.4. All printer output goes to unit 6 as specified in the main program. If necessary, the unit number for printer output can be changed to meet the requirements of a particular computer.

FOR EACH SIMULATION

### BAS1RP

1.	Data:	IHEDFM	IDDNFM	IHEDUN	IDDNUN
	Format:	I10	I10	I10	I10

FOR EACH TIME STEP

### BAS10C

2.	Data:	INCODE	IHDDFL	IBUDFL	ICBCFL
	Format:	I10	I10	I10	I10
3.	Data:	Hdpr	Ddpr	Hdsv	Ddsv
	Format:	I10	I10	I10	I10

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

### Explanation of Fields Used in Input Instructions

IHEDFM--is a code for the format in which heads will be printed.

IDDNFM--is a code for the format in which drawdowns will be printed. Format codes have the same meaning for both head and drawdown. A positive format code indicates that each row of data is printed completely before starting the next row. This means that when there are more columns in a row than will fit on one line, additional lines are used as required to complete the row. This format is called the wrap format. A negative format code indicates that the printout is broken into strips where only that number of columns that will fit across one line are printed in a strip. As many strips are used as are required to print the entire model width. This format is called the strip format. The absolute value of the format code specifies the printout format as follows.

0 - (10G11.4)	7 - (20F5.0)
1 - (11G10.3)	8 - (20F5.1)
2 - (9G13.6)	9 - (20F5.2)
3 - (15F7.1)	10 - (20F5.3)
4 - (15F7.2)	11 - (20F5.4)
5 - (15F7.3)	12 - (10G11.4)
6 - (15F7.4)	

IHEDUN--is the unit number to which heads will be written if they are saved on disk.

IDDNUN--is the unit number to which drawdowns will be written if they are saved on disk.

INCODE--is the head/drawdown output code. It determines the number of records in input item 3.

If INCODE < 0, layer-by-layer specifications from the last time steps are used. Input item 3 is not read.

If INCODE = 0, all layers are treated the same way. Input item 3 will consist of one record.

If INCODE > 0, input item 3 will consist of one record for each layer.

IHDDFL--is a head and drawdown output flag.

If IHDDFL = 0, neither heads nor drawdowns will be printed or saved on disk.

If IHDDFL ≠ 0, heads and drawdowns will be printed or saved according to the flags for each layer specified in input item 3.

IBUDFL--is a budget print flag.

If IBUDFL = 0, overall volumetric budget will not be printed.

If IBUDFL ≠ 0, overall volumetric budget will be printed.

(Note that the overall volumetric budget will always be printed at the end of a stress period, even if the value of IBUDFL is zero.)

ICBCFL--is a cell-by-cell flow-term flag.

If ICBCFL = 0, cell-by-cell flow terms are not saved or printed.

If ICBCFL ≠ 0, cell-by-cell flow terms are printed or recorded on disk depending on flags set in the component of flow packages, i.e., IWELCB, IRCHCB, etc.

Hdpr--is the output flag for head printout.

If  $Hdpr = 0$ , head is not printed for the corresponding layer.

If  $Hdpr \neq 0$ , head is printed for the corresponding layer.

Ddpr--is the output flag for drawdown printout.

If  $Ddpr = 0$ , drawdown is not printed for the corresponding layer.

If  $Ddpr \neq 0$ , drawdown is printed for the corresponding layer.

Hdsv--is the output flag for head save.

If  $Hdsv = 0$ , head is not saved for the corresponding layer.

If  $Hdsv \neq 0$ , head is saved for the corresponding layer.

Ddsv--is the output flag for drawdown save.

If  $Ddsv = 0$ , drawdown is not saved for the corresponding layer.

If  $Ddsv \neq 0$ , drawdown is saved for the corresponding layer.

SAMPLE INPUT TO THE OUTPUT CONTROL OPTION

INPUT RECORDS

EXPLANATION

DATA  
ITEM

DATA ITEM	EXPLANATION	4	8	76	77
1	{IHEDFM, IDDNFM, IHEDUN, IDDNUN}	1	1	0	0
2	TIME STEP 1--{INCODE, IHDDFL, IBUDEF, ICBCFL}	1	1	1	1
3	LAYER 1--{HDPR, DDPR, HDSV, DDSV}	1	1	0	0
3	LAYER 2--{HDPR, DDPR, HDSV, DDSV}	1	1	0	0
3	LAYER 3--{HDPR, DDPR, HDSV, DDSV}	1	1	0	0
2	TIME STEP 2--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	0	1	0
2	TIME STEP 3--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	1	1	0
2	TIME STEP 4--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	0	1	0
2	TIME STEP 5--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	1	1	1
2	TIME STEP 6--{INCODE, IHDDFL, IBUDEF, ICBCFL}	0	1	1	0
3	ALL LAYERS--{HDPR, DDPR, HDSV, DDSV}	1	1	0	0
2	TIME STEP 7--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	0	1	0
2	TIME STEP 8--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	1	1	0
2	TIME STEP 9--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	0	1	0
2	TIME STEP 10--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	1	1	0
2	TIME STEP 11--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	0	1	0
2	TIME STEP 12--{INCODE, IHDDFL, IBUDEF, ICBCFL}	-1	1	1	1

## 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 initializes heads at the beginning of each time step.
BAS1FM	Clears accumulators RHS and HCOF.
BAS10C	Sets flags which indicate when data should be printed or recorded on disk.
BAS10T	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.

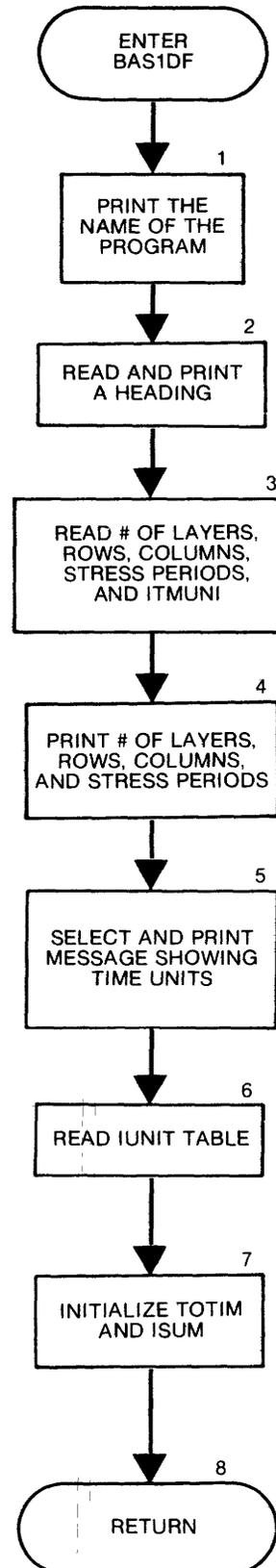
## Flow Chart for Module BAS1DF

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.



```

SUBROUTINE BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,
1      NLAY,NODES,INBAS,IOUT,IUNIT)
C
C-----VERSION 1128 28DEC1983 BAS1DF
C *****
C DEFINE KEY MODEL PARAMETERS
C *****
C
C SPECIFICATIONS:
C -----
C 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

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
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 BASIAL

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

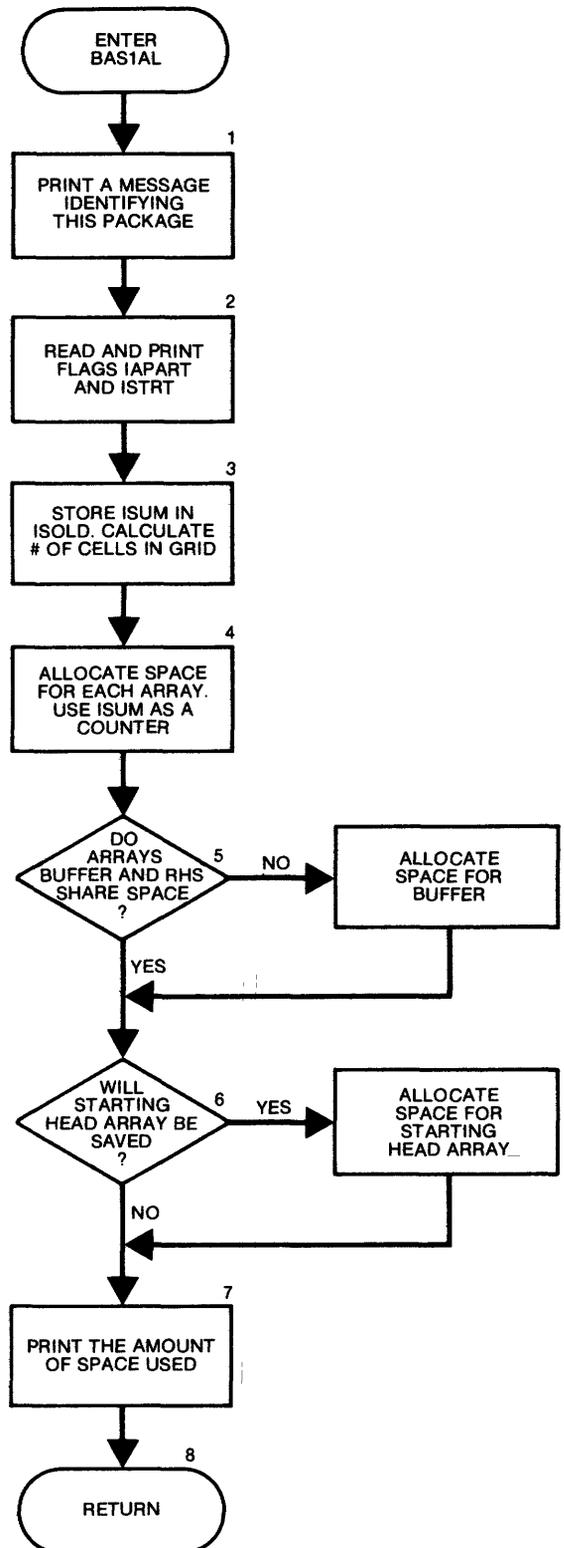
BUFFER is an array in which data is temporarily stored while it is being gathered for printing.

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 BASIAL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
1          LCHCOF,LCRHS,LCDELR,LCDELCLCSTRT,LCBUFF,LCIOFL,INBAS,
1          ISTRT,NCOL,NROW,NLAY,IOUT)
C-----VERSION 0927 08DEC1983 BASIAL
C *****
C ALLOCATE SPACE FOR BASIC MODEL ARRAYS
C *****
C SPECIFICATIONS:
C -----
C -----
C1-----PRINT A MESSAGE IDENTIFYING THE PACKAGE.
WRITE(IOUT,1)INBAS
1 FORMAT(1H0,'BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 12/08/83',
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
LCDELCLC=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,I6,' ELEMENTS IN X ARRAY ARE USED BY BAS')
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
C8-----RETURN
RETURN
C
END

```

List of Variables for Module BASIAL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
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.
ISTR	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.
LCDEL	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.
LHCOF	Package	Location in the X array of the first element of array HCOF.
LHNEW	Package	Location in the X array of the first element of array HNEW.
LHOLD	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.
LCSTR	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.

<u>Code</u>	<u>Status</u>
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

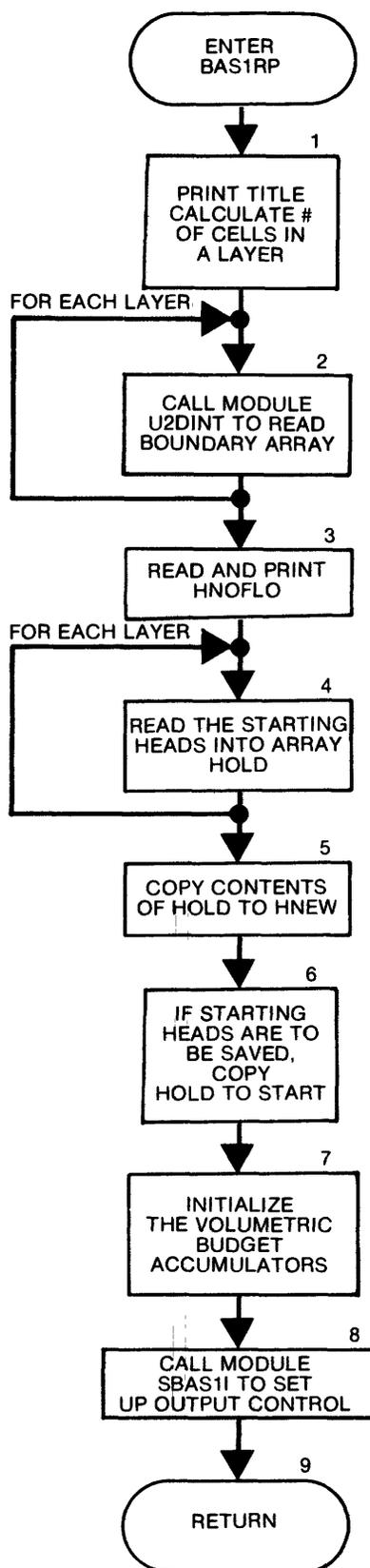
HNOFLO is a value assigned to head in inactive (no-flow) cells. It makes those cells stand out in listings of heads.

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.





List of Variables for Module BAS1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
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.
IHFDM	Package	Code for format in which head should be printed.
IHDUN	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,4) ≠ 0, drawdown will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISTR	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.
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

$$\text{DELT} = (1 - \text{TSMULT}) * \text{PERLEN} / (1 - \text{TSMULT} ** \text{NSTP}).$$

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.

## Flow Chart for Module BAS1ST

PERLEN is the length of a stress period.

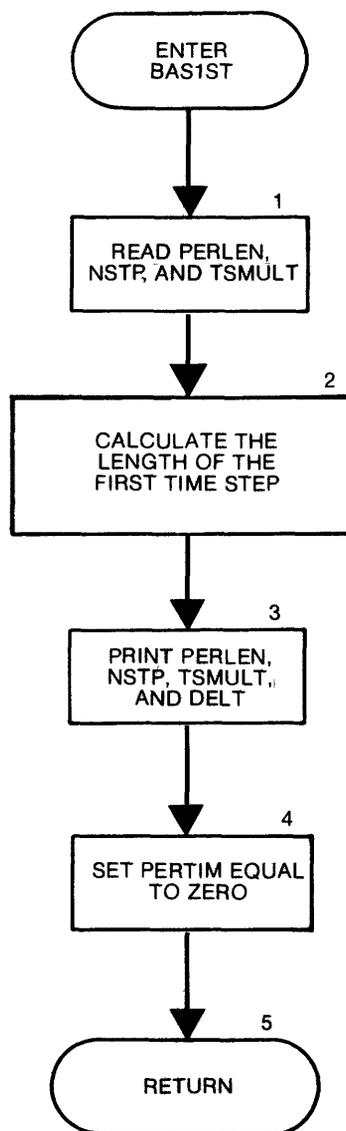
NSTP is the number of time steps in a stress period.

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:

$$\text{DELT} = (1 - \text{TSMULT}) * \text{PERLEN} / (1 - \text{TSMULT} ** \text{NSTP})$$

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 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

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
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 (DELTA). 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.

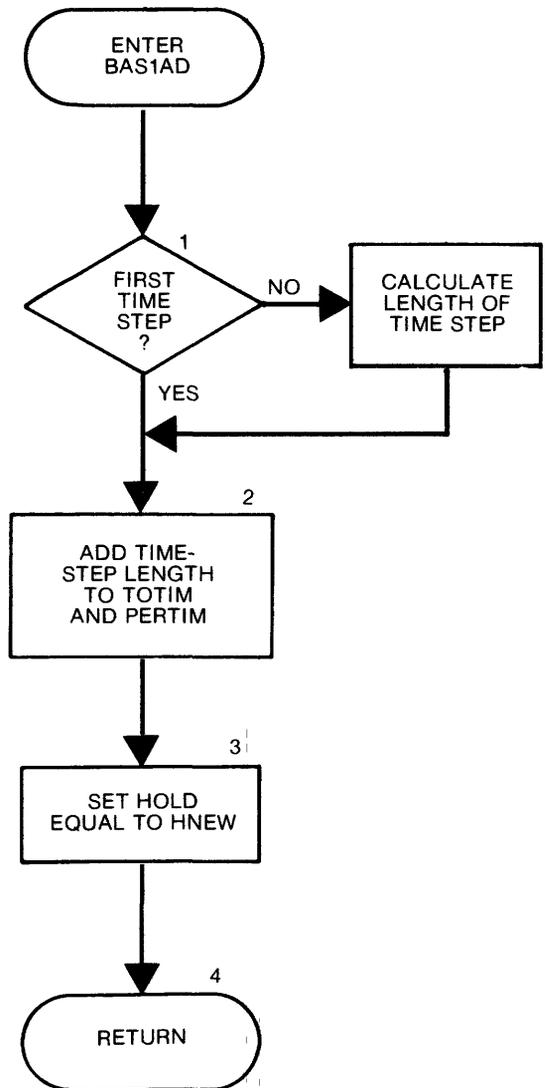
## Flow Chart for Module BAS1AD

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.



```

      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 -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY), HOLD(NCOL,NROW,NLAY)
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

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
DELTA	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,NCOL,NROW,NLAY,NODES)
C
C *****
C SET HCOF=RHS=0.
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION HCOF(NODES),RHS(NODES)
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

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
I	Module	Index.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in cell (J,I,K) in the finite-difference equation.
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.
NROW	Global	Number of rows in the grid.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

## Narrative for Module BAS10C

Module BAS10C 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 BAS10C 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 (BAS10C) 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 BAS10C

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:

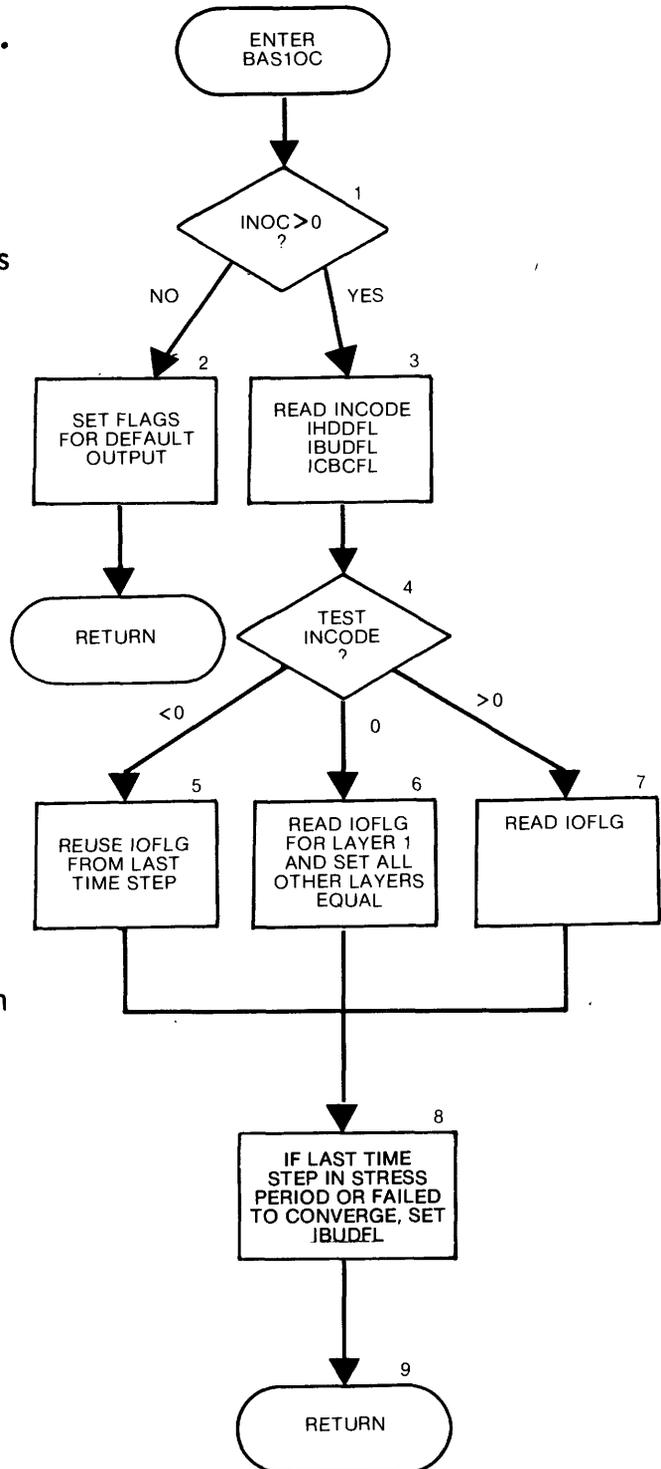
- 1---head print
- 2---drawdown print
- 3---head save
- 4---drawdown save

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 BASIOC(NSTP,KSTP,KPER,ISTR,ICNVG,IOFLG,NLAY,
1      IBUDFL,ICBCFL,IHDDFL,INOC,IOUT)
C
C-----VERSION 0949 03NOV1982 BASIOC
C *****
C      OUTPUT CONTROLLER FOR HEAD, DRAWDOWN, AND BUDGET
C      *****
C
C      SPECIFICATIONS:
C      -----
C      DIMENSION IOFLG(NLAY,4)
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 BAS10C

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
I IBUDFL	Module Package	Index. 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 ISTR	Global Package	Primary unit number for all printed output. IOUT = 6. Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
K	Module	Layer index.
KPER	Global	Stress period counter.
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.

## Narrative for Module BAS10T

Module BAS10T invokes submodules which write results of the simulation. Those results include head, drawdown, overall volumetric budget, and a time summary. Results are printed according to flags IHDDFL, IOFLG, and IBUDFL which are set by module BAS10C (Output Control). If flag IHDDFL is set, a table of flags named IOFLG is used to determine which heads and drawdown should be written (printer or disk) and for which layers it should be written. This module (BAS10T) calls submodules SBAS1H and SBAS1D to write heads and drawdowns respectively. If flag IBUDFL is set, submodule SBAS1V is invoked to calculate and print the overall volumetric budget. After every time step during which results have been printed, a time summary is printed.

Module BAS10T performs its functions in the following order:

1. Clear flag IPFLG. This flag is set later in this module if any results are printed. It controls the printing of a time summary.
2. If the iterative procedure failed to converge, print a message to that effect.
3. If the head and drawdown flag (IHDDFL) are set, call submodules SBAS1H and SBAS1D to write heads and drawdowns in accordance with the flags in the table IOFLG.
4. If the budget flag (IBUDFL) is set, call submodule SBAS1V to calculate and print the volumetric budget.
5. If the printout flag (IPFLG) is set, call submodule SBAS1T to print a time summary.
6. RETURN.

Flow Chart for Module BAS10T

IPFLG is the printout flag. It is set when any results are printed. If it is set, a time summary is printed.

SBAS1H is a submodule which writes heads.

SBAS1D is a submodule which writes drawdown.

SBAS1V is a submodule which prints the volumetric budget.

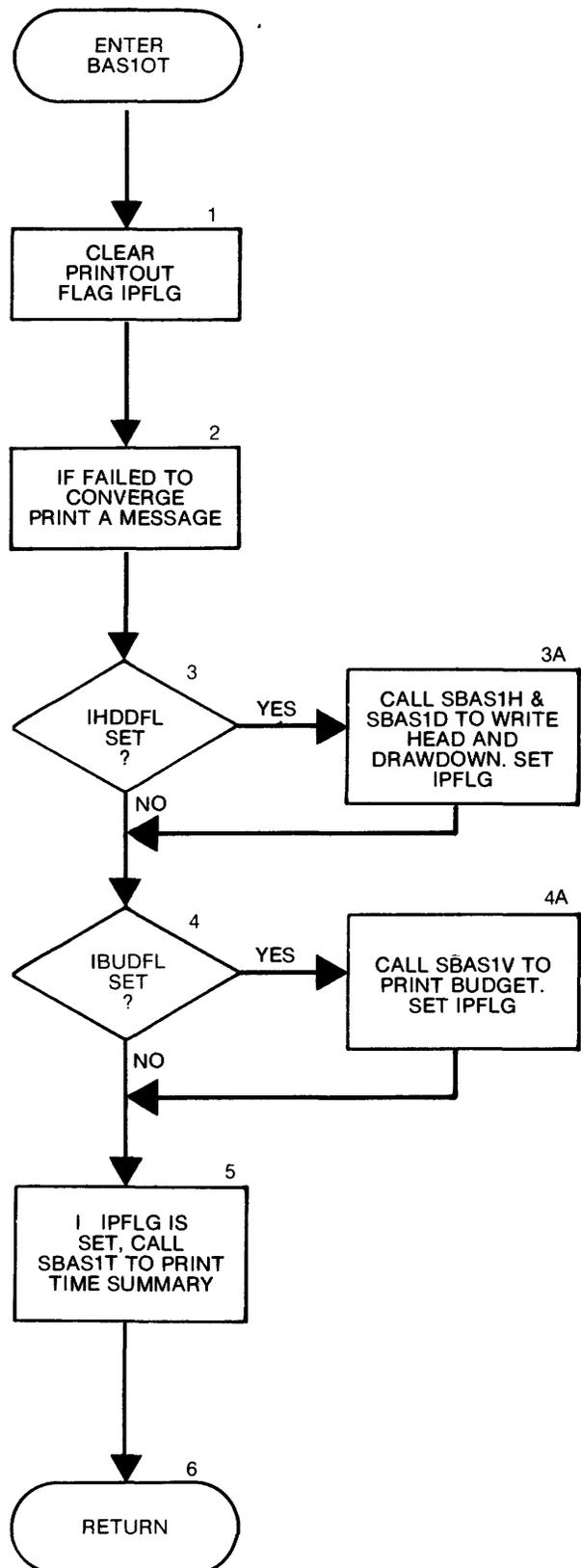
SBAS1T is a submodule which prints a time summary.

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

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

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



```

SUBROUTINE BAS10T(HNEW,STRT,ISTR,IBUFF,IOFLG,MSUM,IBOUND,VBNM,
1  VBVL,KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
2  IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
C-----VERSION 1154 29MAR1984 BAS10T
C *****
C OUTPUT TIME, VOLUMETRIC BUDGET, HEAD, AND DRAWDOWN
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
1  VBVM(1),VBVL(1),IOFLG(NLAY,4),
2  IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
C -----
C
C1-----CLEAR PRINTOUT FLAG (IPFLG)
IPFLG=0
C
C2-----IF ITERATIVE PROCEDURE FAILED TO CONVERGE PRINT MESSAGE
IF(ICNVG.EQ.0) WRITE(IOUT,1) KSTP,KPER
1  FORMAT(1H0,10X,'****FAILED TO CONVERGE IN TIME STEP',I3,
1  ' OF STRESS PERIOD',I3,'****')
C
C3-----IF HEAD AND DRAWDOWN FLAG (IHDDFL) IS SET WRITE HEAD AND
C3-----DRAWDOWN IN ACCORDANCE WITH FLAGS IN IOFLG.
IF(IHDDFL.EQ.0) GO TO 100
C
CALL SBAS1H(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
1  NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
CALL SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,NLAY,IOUT,
1  IDDNFM,IDDNUN,STRT,ISTR,IBOUND,IPFLG,PERTIM,TOTIM)
C
C4-----PRINT TOTAL BUDGET IF REQUESTED
100 IF(IBUDFL.EQ.0) GO TO 120
CALL SBAS1V(MSUM,VBVM,VBVL,KSTP,KPER,IOUT)
IPFLG=1
C
C5-----END PRINTOUT WITH TIME SUMMARY AND FORM FEED IF ANY PRINTOUT
C5-----WILL BE PRODUCED.
120 IF(IPFLG.EQ.0) RETURN
CALL SBAS1T(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)
WRITE(IOUT,101)
101 FORMAT(1H1)
C
C6-----RETURN
RETURN
END

```

List of Variables for Module BAS10T

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
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.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
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.
ICNVG	Global	Flag is set equal to one when the iteration procedure has converged.
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.
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.
IHEDFM	Package	Code for the format in which head should be printed.
IHEDUN	Package	Unit number on which an unformatted record containing head should be recorded.
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.
IPFLG	Package	Flag. = 0 means nothing has been printed this time step. ≠ 0 means something has been printed this time step; therefore, a time summary must be printed.
ISTRT	Package	Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
ITMUNI	Package	Code for time units for this problem: 0 - undefined 1 - seconds 2 - minutes 3 - hours 4 - days 5 - years

List of Variables for Module BAS10T (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
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.
STRT	Package	DIMENSION (NCOL,NROW,NLAY), Starting head.
TOTIM	Package	Elapsed time in the simulation.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N) Rate for the current time step into the flow field. (2,N) Rate for the 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 SBAS1D

Module SBAS1D is called by module BAS10T to calculate and write drawdown for every cell in certain layers in the grid. The module is called at the end of each time step if the head and drawdown flag (IHDDFL) is set. It calculates drawdown only if the user has specified that starting heads should be saved.

The layers for which drawdown is to be written are determined by the settings of flags in the table named IOFLG. In IOFLG, there are four flags for each layer. The second flag, if it is set, causes drawdown to be printed. The fourth flag, if it is set, causes drawdown to be recorded.

Module SBAS1D performs its functions in the following order:

1. For each layer, do steps 2-5.
2. If flags indicate that drawdown is not needed for this layer, go on to the next layer.
3. Test flag ISTRT to see if starting heads were saved. Go to either 4 or 5.
4. Starting heads were not saved. Write a message to that effect and STOP.
5. Starting heads were saved. Calculate drawdown for this layer.
6. For each layer, if drawdown is to be printed, call module ULAPRS or ULAPRW, depending on the format requested (IDDNFM), to print drawdown.
7. For each layer, if drawdown is to be recorded, call module ULASAV to write the drawdown to the unit specified in IDDNUN.
8. RETURN.

## Flow Chart for Module SBAS1D

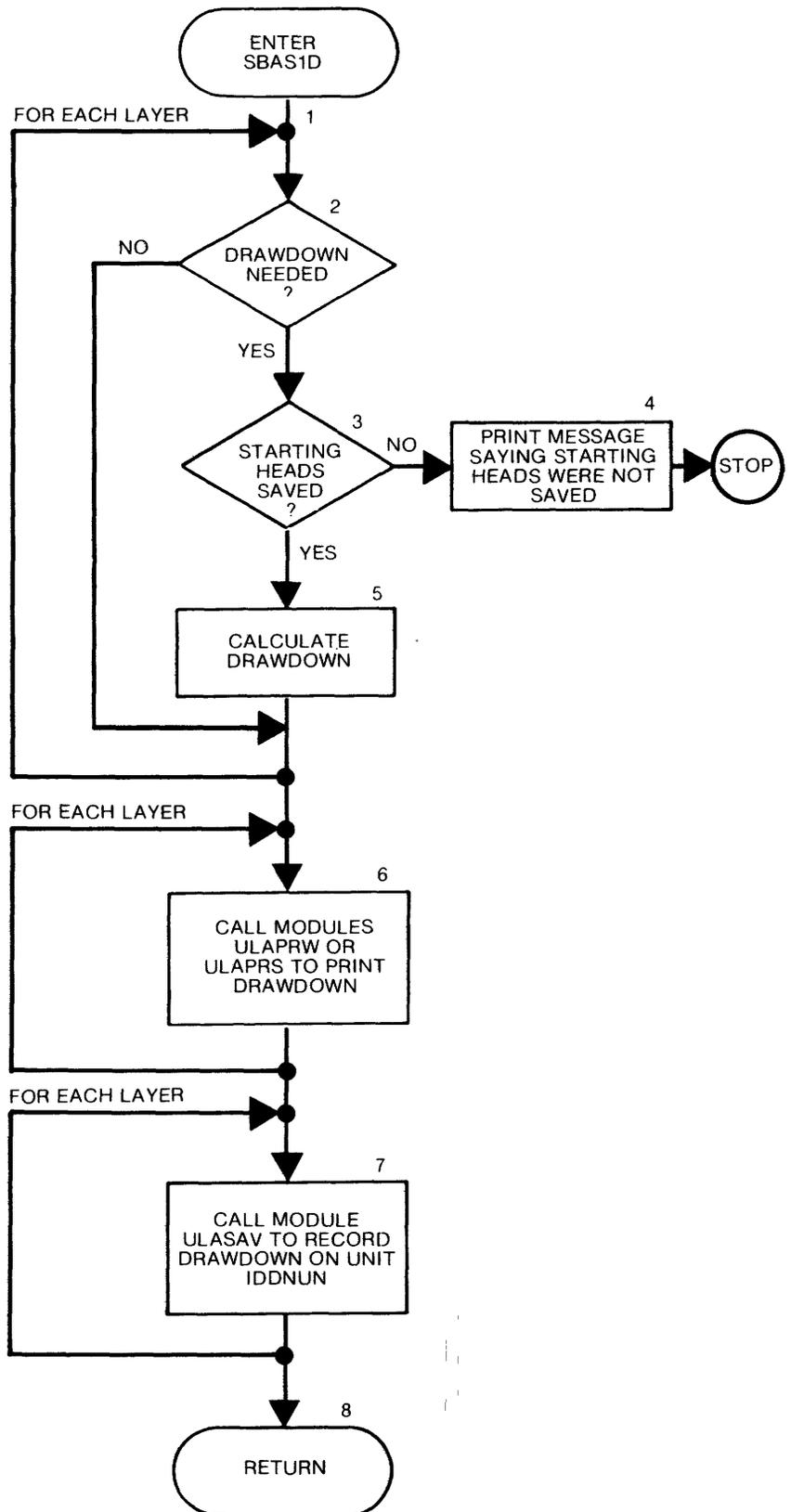
IOFLG is a table containing one entry for each layer. Each element consists of four flags which, when set, cause (1) head to be printed, (2) drawdown to be printed, (3) head to be recorded, and (4) drawdown to be recorded.

ULAPRW is a utility module which prints a value for each cell in a wrap format. In the wrap format, all values for one row are printed before any values for the next row.

ULAPRS is a utility module which prints a value for each cell in the layer in a strip format. In the strip format, all values in a group of N columns are printed before any values in the next N columns are printed.

ULASAV is a utility module which records a value for each cell in a layer.

IDDNUN is a unit number, specified by the user, on which drawdown will be recorded.



```

        SUBROUTINE SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
1         NLAY,IOUT,IDDNFM,IDDNUN,STRT,ISTR,IBOUND,IPFLG,
2         PERTIM,TOTIM)
C
C-----VERSION 1147 29MAR1984 SBAS1D
C *****:*****
C CALCULATE PRINT AND RECORD DRAWDOWNS
C *****:*****
C
C SPECIFICATIONS
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),TEXT(4),
1         BUFF(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
2         IBOUND(NCOL,NROW,NLAY)
C
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' ',' ','DRAW',
1         'DOWN'/
C -----
C
C1-----FOR EACH LAYER CALCULATE DRAWDOWN IF PRINT OR RECORD
C1-----IS REQUESTED
C         DO 59 K=1,NLAY
C
C2-----IS DRAWDOWN NEEDED FRO THIS LAYER?
C         IF(IOFLG(K,2).EQ.0 .AND. IOFLG(K,4).EQ.0) GO TO 59
C
C3-----DRAWDOWN IS NEEDED. WERE STARTING HEADS SAVED?
C         IF(ISTR.NE.0) GO TO 53
C
C4-----STARTING HEADS WERE NOT SAVED. PRINT MESSAGE AND STOP.
C         WRITE(IOUT,52)
C         52 FORMAT(1HO,'CANNOT CALCULATE DRAWDOWN BECAUSE START',
1         ' HEADS WERE NOT SAVED')
C         STOP
C
C5-----CALCULATE DRAWDOWN FOR THE LAYER.
C         53 DO 58 I=1,NROW
C             DO 58 J=1,NCOL
C                 HSING=HNEW(J,I,K)
C                 BUFF(J,I,K)=HSING
C                 IF(IBOUND(J,I,K).NE.0) BUFF(J,I,K)=STRT(J,I,K)-HSING
C             58 CONTINUE
C         59 CONTINUE
C
C6-----FOR EACH LAYER: DETERMINE IF DRAWDOWN SHOULD BE PRINTED.
C6-----IF SO THEN CALL ULAPRS OR ULAPRW TO PRINT DRAWDOWN.
C         DO 69 K=1,NLAY
C             IF(IOFLG(K,2).EQ.0) GO TO 69
C             IF(IDDNFM.LT.0) CALL ULAPRS(BUFF(1,1,K),TEXT(1),KSTP,KPER,
1             NCOL,NROW,K,-IDDNFM,IOUT)
C             IF(IDDNFM.GE.0) CALL ULAPRW(BUFF(1,1,K),TEXT(1),KSTP,KPER,
1             NCOL,NROW,K,IDDNFM,IOUT)
C             IPFLG=1
C         69 CONTINUE
C
C7-----FOR EACH LAYER: DETERMINE IF DRAWDOWN SHOULD BE RECORDED.
C7-----IF SO THEN CALL ULASAV TO RECORD DRAWDOWN.
C         IFIRST=1
C         IF(IDDNUN.LE.0) GO TO 80
C         DO 79 K=1,NLAY
C             IF(IOFLG(K,4).LE.0) GO TO 79
C             IF(IFIRST.EQ.1) WRITE(IOUT,74) IDDNUN,KSTP,KPER
C             74 FORMAT(1HO,'DRAWDOWN WILL BE SAVED ON UNIT',I3,
1             ' AT END OF TIME STEP',I3,' , STRESS PERIOD',I3)
C             IFIRST=0
C             CALL ULASAV(BUFF(1,1,K),TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,
1             NROW,K,IDDNUN)
C         79 CONTINUE
C
C8-----RETURN
C         80 RETURN
C         END

```

List of Variables for Module SBAS1D

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HSING	Module	Single-precision temporary field for HNEW (J,I,K).
I	Module	Index for rows.
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.
IFIRST	Module	Flag to indicate that a notice should be printed when drawdown is recorded.
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.
IPFLG	Package	Flag. = 0 means nothing has been printed this time step. ≠ 0 means something has been printed this time step; therefore, a time summary must be printed.
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 for columns.
K	Module	Index for layers.
KPER	Global	Stress period counter.
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.
STRT	Package	DIMENSION (NCOL,NROW,NLAY), Starting head.
TEXT	Module	Label to be printed or recorded with array data.
TOTIM	Package	Elapsed time in the simulation.

## Narrative for Module SBAS1H

Module SBAS1H prints and records head for every cell in certain layers in the grid. It is called by module BAS10T at the end of each time step if the head and drawdown flag (IHDDFL) is set. The layers for which head is written is controlled by the settings of flags in the table named IOFLG. In IOFLG, there are four flags for each layer. The first flag, if it is set, causes head for the corresponding layer to be printed. The third flag, if it is set, causes head to be recorded.

Module SBAS1H performs its functions in the following order:

1. For each layer, DO STEPS 2-4.
2. Test the flag table (IOFLG) to see if heads should be printed for this layer. If so, DO STEPS 3 AND 4.
3. Copy heads for this layer (which are contained in the double-precision array (HNEW)) into the single-precision buffer array (BUFF).
4. Depending on the print-format code, call either module ULAPRW or ULAPRS to print the contents of the buffer array.
5. Test the unit number for recording heads (IHEDUN) to see if it is positive. If it is not positive, heads will not be recorded (SKIP STEPS 6-9). If it is positive, heads may be recorded in accordance with the setting of flags in the IOFLG array. DO STEPS 6-9.
6. For each layer, DO STEPS 7-9.
7. If flags in IOFLG indicate that heads are not to be recorded for this layer, move on to the next layer.
8. Copy heads from the HNEW array (double-precision) to the BUFF array (single-precision).
9. Call module ULASAV to record the heads on unit IHEDUN.
10. RETURN.

## Flow Chart for Module SBAS1H

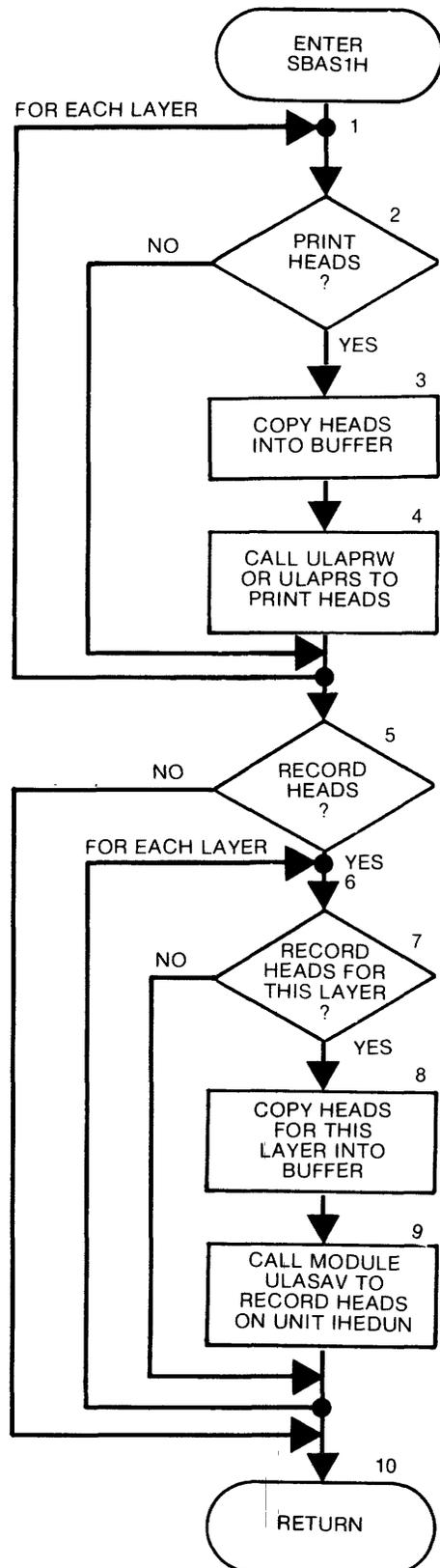
IOFLG is a table containing one entry for each layer. Each element consists of four flags which, when set, cause (1) head to be printed, (2) drawdown to be printed, (3) head to be recorded, and (4) drawdown to be recorded.

ULAPRW is a utility module which prints a value for each cell in the layer in a wrap format. In the wrap format, all values for one row are printed before any values for the next row.

ULAPRS is a utility module which prints a value for each cell in the layer in a strip format. In the strip format, all values in a group of N columns are printed before any values in the next N columns are printed.

ULASAV is a utility module which records a value for each cell in a layer.

IHEDUN is a unit number, specified by the user, on which heads will be recorded. If IHEDUN is less than or equal to zero, heads will not be recorded.



```

SUBROUTINE SBAS1H(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
1  NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
C
C-----VERSION 1138 29MAR1984 SBAS1H
C *****
C PRINT AND RECORD HEADS
C *****
C
C SPECIFICATIONS
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),TEXT(4),
1  BUFF(NCOL,NROW,NLAY)
C
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' ',',',',',',
1  'HEAD'/
C -----
C1-----FOR EACH LAYER: PRINT HEAD IF REQUESTED.
DO 39 K=1,NLAY
C
C2-----TEST IOFLG TO SEE IF HEAD SHOULD BE PRINTED.
IF(IOFLG(K,1).EQ.0) GO TO 39
IPFLG=1
C
C3-----COPY HEADS FOR THIS LAYER INTO BUFFER.
DO 32 I=1,NROW
DO 32 J=1,NCOL
BUFF(J,I,1)=HNEW(J,I,K)
32 CONTINUE
C
C4-----CALL UTILITY MODULE TO PRINT CONTENTS OF BUFFER.
IF(IHEDFM.LT.0) CALL ULAPRS(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
1  -IHEDFM,IOUT)
IF(IHEDFM.GE.0) CALL ULAPRW(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,K,
1  IHEDFM,IOUT)
39 CONTINUE
C
C5-----IF UNIT FOR RECORDING HEADS <= 0: THEN RETURN.
IF(IHEDUN.LE.0)GO TO 50
IFIRST=1
C
C6-----FOR EACH LAYER: RECORD HEAD IF REQUESTED.
DO 49 K=1,NLAY
C
C7-----CHECK IOFLG TO SEE IF HEAD FOR THIS LAYER SHOULD BE RECORDED.
IF(IOFLG(K,3).LE.0) GO TO 49
IF(IFIRST.EQ.1) WRITE(IOUT,41) IHEDUN,KSTP,KPER
41 FORMAT(1H0,'HEAD WILL BE SAVED ON UNIT',I3,' AT END OF TIME STEP',
1  I3,', STRESS PERIOD',I3)
IFIRST=0
C
C8-----COPY HEADS FOR THIS LAYER INTO BUFFER.
DO 44 I=1,NROW
DO 44 J=1,NCOL
BUFF(J,I,1)=HNEW(J,I,K)
44 CONTINUE
C
C9-----RECORD CONTENTS OF BUFFER ON UNIT=IHEDUN
CALL ULASAV(BUFF,TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,NROW,K,
1  IHEDUN)
49 CONTINUE
C
C10-----RETURN
50 RETURN
END

```

List of Variables for Module SBAS1H

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
I	Module	Index for rows.
IFIRST	Module	Flag which, if set (equal to 1), indicates that a notice should be printed when head is 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.
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.
IPFLG	Package	Flag. = 0 means nothing has been printed this time step. ≠ 0 means something has been printed this time step; therefore, a time summary must be printed.
J	Module	Index for columns.
K	Module	Index for layers.
KPER	Global	Stress period counter.
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.
TEXT	Module	Label to be printed or recorded with array data.
TOTIM	Package	Elapsed time in the simulation.

## Narrative for Module SBAS1I

Module SBAS1I initializes the Output Control System. If the user does not opt to control output, the formats for printing head and drawdown are set to the default format and flags are set so that, whenever heads or drawdowns are printed, they are printed for all layers. If the user does opt to control output, the formats for printing and the unit numbers for recording head and drawdown are read.

A table named IOFLG contains one entry for each layer in the grid. Each entry consists of four flags corresponding to four operations: (1) head print, (2) drawdown print, (3) head record, and (4) drawdown record. The module BAS1OT examines the table and, for each layer, performs only the operations for which the corresponding flags are set (equal to one). This module (SBAS1I) sets the head-print flag if the user opts for default output. If starting heads are saved, it also sets the drawdown-print flag. If the user opts to control output, the flags in IOFLG are read at each time step.

Module SBAS1I performs its functions in the following order:

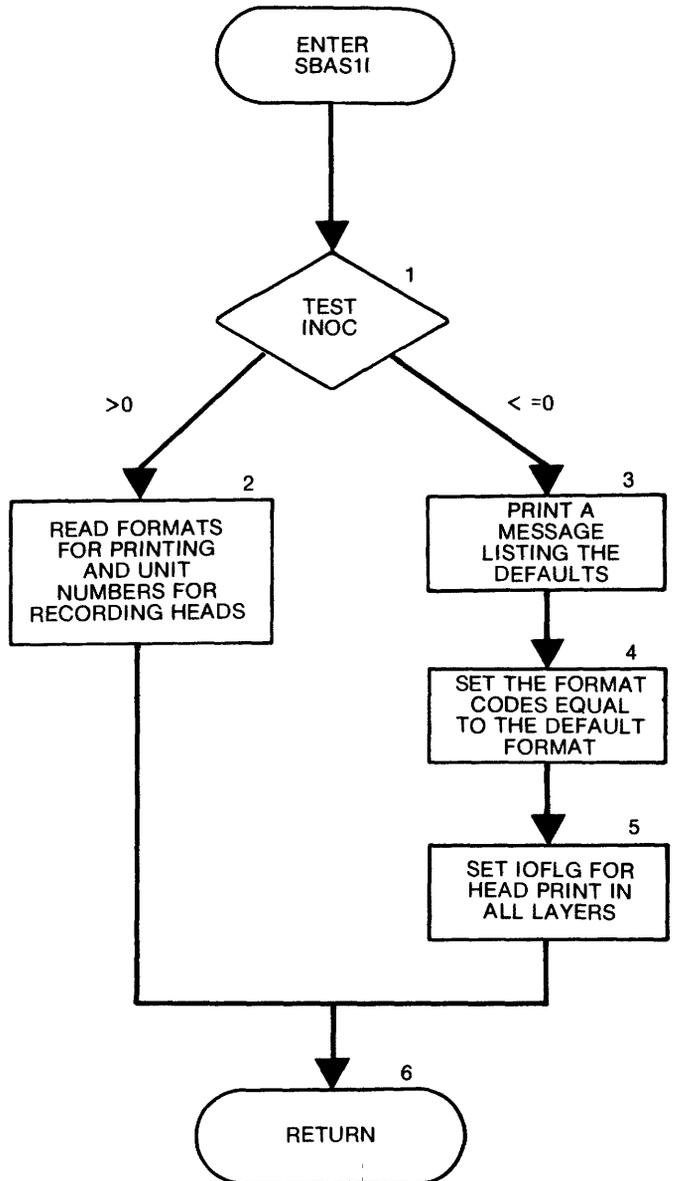
1. Test the unit number for Output Control (IUNIT (12)), which is known in this module by the name INOC, to see if it is positive. If it is positive, the Output Control option is active and output specification will be read from the unit number contained in INOC. If it is not positive, the Output Control option is not active and flags are set to defaults. GO TO 2 OR 3.
2. Output Control is active. Read and print the head-print format code (IHEDFM), the drawdown-print format code (IDDNFM), the unit number to record heads (IHEDUN), and the unit number to record drawdown (IDDNUN). GO TO 6. Note: The formats and associated codes are listed in the Input Instructions for Output Control.
3. Output Control is inactive. Print a message listing the defaults.
4. Set the print-format codes (IHEDFM and IDDNFM) equal to zero to get the default format.
5. Set the flags in IOFLG so that head and drawdown are printed for all layers.
6. RETURN.

## Flow Chart for Module SBAS1I

INOC is the input unit for Output Control. It is the same as element 12 in the IUNIT table. When it is greater than zero, Output Control is active--the user will provide output specifications. When it is less than or equal to zero, Output Control is inactive--output will be controlled by default.

IOFLG is a table containing one entry for each layer. Each entry consists of four flags which, when set, cause

- (1) head to be printed,
- (2) drawdown to be printed,
- (3) head to be recorded, and
- (4) drawdown to be recorded.



```

SUBROUTINE SBAS1(NLAY,ISTR,IOFLG,INOC,IOUT,IHEDFM,
1          IDDNFM,IHEDUN,IDDNUN)
C
C-----VERSION 1138 03NOV1982 SBAS1
C *****
C SET UP OUTPUT CONTROL
C *****
C
C SPECIFICATIONS
C
C DIMENSION IOFLG(NLAY,4)
C -----
C1-----TEST UNIT NUMBER FROM IUNIT (INOC) TO SEE IF OUTPUT
C1-----CONTROL IS ACTIVE.
          IF(INOC.LE.0) GO TO 600
C
C2-----READ AND PRINT FORMATS FOR PRINTING AND UNIT NUMBERS FOR
C2-----RECORDING HEADS AND DRAWDOWN. THEN RETURN.
          500 READ (INOC,1)IHEDFM,IDDNFM,IHEDUN,IDDNUN
              1 FORMAT (4I10)
              WRITE (IOUT,3)IHEDFM,IDDNFM
              3 FORMAT (1H0,'HEAD PRINT FORMAT IS FORMAT NUMBER',I4,
              1 ' DRAWDOWN PRINT FORMAT IS FORMAT NUMBER',I4)
              WRITE (IOUT,4)IHEDUN,IDDNUN
              4 FORMAT (1H0,'HEADS WILL BE SAVED ON UNIT',I3,
              1 ' DRAWDOWNS WILL BE SAVED ON UNIT',I3)
              WRITE(IOUT,561)
          561 FORMAT(1H0,'OUTPUT CONTROL IS SPECIFIED EVERY TIME STEP')
              GO TO 1000
C
C3-----OUTPUT CONTROL IS INACTIVE. PRINT A MESSAGE LISTING DEFAULTS.
          600 WRITE(IOUT,641)
          641 FORMAT(1H0,'DEFAULT OUTPUT CONTROL -- THE FOLLOWING OUTPUT',
          1 ' COMES AT THE END OF EACH STRESS PERIOD:')
              WRITE(IOUT,642)
          642 FORMAT(1X,'TOTAL VOLUMETRIC BUDGET')
              WRITE(IOUT,643)
          643 FORMAT(1X,10X,'HEAD')
              IF(ISTR.NE.0)WRITE(IOUT,644)
          644 FORMAT(1X,10X,'DRAWDOWN')
C
C4-----SET THE FORMAT CODES EQUAL TO THE DEFAULT FORMAT.
          IHEDFM=0
          IDDNFM=0
C
C5-----SET DEFAULT FLAGS IN IOFLG SO THAT HEAD (AND DRAWDOWN) IS
C5-----PRINTED FOR EVERY LAYER.
          ID=0
          IF(ISTR.NE.0) ID=1
          670 DO 680 K=1,NLAY
              IOFLG(K,1)=1
              IOFLG(K,2)=ID
              IOFLG(K,3)=0
              IOFLG(K,4)=0
          680 CONTINUE
              GO TO 1000
C
C6-----RETURN
          1000 RETURN
          END

```

List of Variables for Module SBAS1I

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ID	Module	Flag to show if STRT was saved (one means yes; zero means no).
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.
INOC	Package	Unit number from which input to output control option will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISTR	Package	Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
K	Module	Index for layers
NLAY	Global	Number of layers in the grid.

## Narrative for Module SBAS1T

Submodule SBAS1T prints a time summary which consists of the time-step length and the elapsed time in seconds, minutes, hours, days, and years. The program can use any consistent set of time units. However, the user is given the option to specify the time units that he is using and the program converts those units to all other convenient units. The user specifies time units (ITMUNI) in module BAS1DF.

1. Use the time-unit indicator (ITMUNI) to determine the conversion factor (CNV) needed to convert time to seconds.

2. If the conversion factor is equal to zero, nonstandard time units are being used.

- (a) Print the time-step length and the elapsed time in the nonstandard units.

- (b) RETURN.

3. Calculate the length of the time step and the elapsed times in seconds.

4. Calculate the time-step length and the elapsed times in minutes, hours, days, and years.

5. Print the time-step length and the elapsed times in all time units.

6. RETURN.

```

SUBROUTINE SBAS1(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)
C
C
C-----VERSION 0837 09APR1982 SBAS1
C *****
C PRINT SIMULATION TIME
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
WRITE(IOUT,199) KSTP,KPER
199 FORMAT(1H0,///10X,'TIME SUMMARY AT END OF TIME STEP',I3,
1 ' IN STRESS PERIOD',I3)
C
C1-----USE TIME UNIT INDICATOR TO GET FACTOR TO CONVERT TO SECONDS.
CNV=0.
IF(ITMUNI.EQ.1) CNV=1.
IF(ITMUNI.EQ.2) CNV=60.
IF(ITMUNI.EQ.3) CNV=3600.
IF(ITMUNI.EQ.4) CNV=86400.
IF(ITMUNI.EQ.5) CNV=31557600.
C
C2-----IF FACTOR=0 THEN TIME UNITS ARE NON-STANDARD.
IF(CNV.NE.0.) GO TO 100
C
C2A-----PRINT TIMES IN NON-STANDARD TIME UNITS.
WRITE(IOUT,301) DELT,PERTIM,TOTIM
301 FORMAT(21X,' TIME STEP LENGTH =',G15.6/
1 21X,' STRESS PERIOD TIME =',G15.6/
2 21X,' TOTAL SIMULATION TIME =',G15.6)
C
C2B-----RETURN
RETURN
C
C3-----CALCULATE LENGTH OF TIME STEP & ELAPSED TIMES IN SECONDS.
100 DELSEC=CNV*DELT
TOTSEC=CNV*TOTIM
PERSEC=CNV*PERTIM
C
C4-----CALCULATE TIMES IN MINUTES,HOURS,DAYS AND YEARS.
DELMN=DELSEC/60.
DELHR=DELMN/60.
DELDY=DELHR/24.
DELYR=DELDY/365.25
TOTMN=TOTSEC/60.
TOTHR=TOTMN/60.
TOTDY=TOTHR/24.
TOTYR=TOTDY/365.25
PERMN=PERSEC/60.
PERHR=PERMN/60.
PERDY=PERHR/24.
PERYR=PERDY/365.25
C
C5-----PRINT TIME STEP LENGTH AND ELAPSED TIMES IN ALL TIME UNITS.
WRITE(IOUT,200)
200 FORMAT(27X,' SECONDS MINUTES HOURS',10X,
1 'DAYS YEARS'/27X,75('-'))
WRITE(IOUT,201) DELSEC,DELMN,DELHR,DELDY,DELYR
201 FORMAT(1X,' TIME STEP LENGTH',5X,5G15.6)
WRITE(IOUT,202) PERSEC,PERMN,PERHR,PERDY,PERYR
202 FORMAT(1X,' STRESS PERIOD TIME',5X,5G15.6)
WRITE(IOUT,203) TOTSEC,TOTMN,TOTHR,TOTDY,TOTYR
203 FORMAT(1X,' TOTAL SIMULATION TIME',5X,5G15.6)
C
C6-----RETURN
RETURN
END

```

List of Variables for Module SBAS1T

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
CNV	Module	Factor to convert elapsed time from units, specified by the user, to seconds.
DELDY	Module	Length of the time step in days.
DELHR	Module	Length of the time step in hours.
DELMN	Module	Length of the time step in minutes.
DELSEC	Module	Length of the time step in seconds.
DELT	Global	Length of the current time step.
DELYR	Module	Length of the time step in years.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMUNI	Package	Code for time units for this problem: 0 - undefined 1 - seconds 2 - minutes 3 - hours 4 - days 5 - years
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
PERDY	Module	Elapsed time in the stress period in days.
PERHR	Module	Elapsed time in the stress period in hours.
PERMN	Module	Elapsed time in the stress period in minutes.
PERSEC	Module	Elapsed time in the stress period in seconds.
PERTIM	Package	Elapsed time during the current stress period.
PERYR	Module	Elapsed time in the stress period in years.
TOTDY	Module	Elapsed time in the simulation in days.
TOTHR	Module	Elapsed time in the simulation in hours.
TOTIM	Package	Elapsed time in the simulation.
TOTMN	Module	Elapsed time in the simulation in minutes.
TOTSEC	Module	Elapsed time in the simulation in seconds.
TOTYR	Module	Elapsed time in the simulation in years.

## Narrative for Module SBAS1V

Module SBAS1V calculates and prints the overall volumetric budget. The individual entries for the budget, which are calculated by the budget modules in each of the component-of-flow packages, are passed to this module in a table named VBVL.

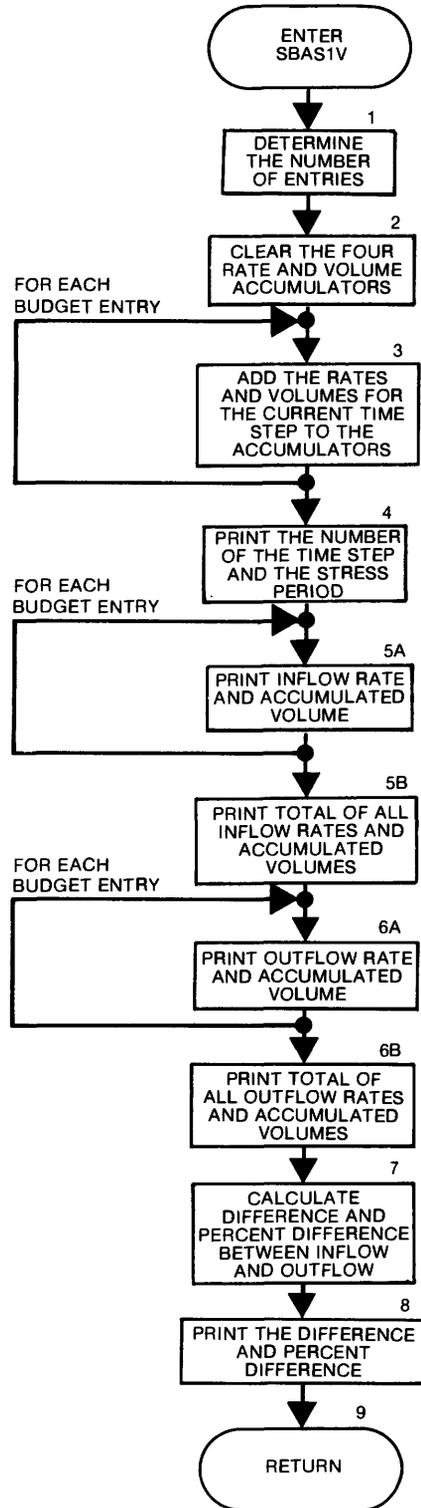
Each entry in VBVL corresponds to a component-of-flow. It consists of four values: rate of inflow for the current time step, rate of outflow for the current time step, accumulated volume of inflow since the beginning of the simulation, and accumulated volume of outflow since the beginning of the simulation. In this module, the total of all inflow rates (TOTRIN), outflow rates (TOTROT), inflow-accumulated volumes (TOTVIN), and outflow-accumulated volumes (TOTVOT) are calculated. The percent differences between those totals are also calculated and printed. The labels for the entries are supplied by the budget modules in the component-of-flow packages and passed in the table VBNM.

Module SBAS1V performs its functions in the following order:

1. Use the counter MSUM to determine the number of individual budget terms (MSUM1).
2. Clear the four accumulators for rates and volumes. The accumulators are total rate into the system (TOTRIN), total rate out of the system (TOTROT), accumulated volume into the system (TOTVIN), and accumulated volume out of the system (TOTVOT).
3. For each source or sink, add the budget entries (rates and volumes), calculated by the budget modules, to the accumulators.
4. Print the number of the time step and stress period.
5. Print the individual input rates and volumes and their totals.
6. Print the individual output rates and volumes and their totals.
7. Calculate the difference between flow into and out of the simulated-flow system. Calculate the percent difference between input and output rates  $(100 * (TOTRIN - TOTROT) / ((TOTRIN + TOTROT) / 2))$ . Calculate the percent difference between input and output accumulated volumes  $(100 * (TOTVIN - TOTVOT) / ((TOTVIN + TOTVOT) / 2))$ .
8. Print the differences and percent differences between input and output rates and volumes.
9. RETURN.

## Flow Chart for Module SBAS1V

The inflow and outflow rates for the current time step and accumulated volumes since the beginning of the simulation for each budget entry (component-of-flow) are added to the four accumulators to obtain the total inflow and outflow rates for the current time step and the total accumulated volume of flow in and volume of flow out since the start of the simulation.



```

SUBROUTINE SBAS1V(MSUM,VBNM,VBVL,KSTP,KPER,IOUT)
C
C
C-----VERSION 1153 03NOV1982 SBAS1V
C *****
C PRINT VOLUMETRIC BUDGET
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION VBNM(4,20),VBVL(4,20)
C -----
C
C1-----DETERMINE NUMBER OF INDIVIDUAL BUDGET ENTRIES.
      MSUM1=MSUM-1
      IF(MSUM1.LE.0) RETURN
C
C2-----CLEAR RATE AND VOLUME ACCUMULATORS.
      TOTRIN=0.
      TOTROT=0.
      TOTVIN=0.
      TOTVOT=0.
C
C3-----ADD RATES AND VOLUMES (IN AND OUT) TO ACCUMULATORS.
      DO 100 L=1,MSUM1
        TOTRIN=TOTRIN+VBVL(3,L)
        TOTROT=TOTROT+VBVL(4,L)
        TOTVIN=TOTVIN+VBVL(1,L)
        TOTVOT=TOTVOT+VBVL(2,L)
      100 CONTINUE
C
C4-----PRINT TIME STEP NUMBER AND STRESS PERIOD NUMBER.
      WRITE(IOUT,260) KSTP,KPER
      WRITE(IOUT,265)
C
C5-----PRINT INDIVIDUAL INFLOW RATES AND VOLUMES AND THEIR TOTALS.
      DO 200 L=1,MSUM1
        WRITE(IOUT,275) (VBNM(I,L),I=1,4),VBVL(1,L),(VBNM(I,L),I=1,4)
        1,VBVL(3,L)
      200 CONTINUE
      WRITE(IOUT,286) TOTVIN,TOTRIN
C
C6-----PRINT INDIVIDUAL OUTFLOW RATES AND VOLUMES AND THEIR TOTALS.
      WRITE(IOUT,287)
      DO 250 L=1,MSUM1
        WRITE(IOUT,275) (VBNM(I,L),I=1,4),VBVL(2,L),(VBNM(I,L),I=1,4)
        1,VBVL(4,L)
      250 CONTINUE
      WRITE(IOUT,298) TOTVOT,TOTROT
C
C7-----CALCULATE THE DIFFERENCE BETWEEN INFLOW AND OUTFLOW.
C
C7A-----CALCULATE DIFFERENCE BETWEEN RATE IN AND RATE OUT.
      DIFFR=TOTRIN-TOTROT

```

```

C
C7B-----CALCULATE PERCENT DIFFERENCE BETWEEN RATE IN AND RATE OUT.
      PDIFFR=100.*DIFFR/((TOTRIN+TOTROT)/2)
C
C7C-----CALCULATE DIFFERENCE BETWEEN VOLUME IN AND VOLUME OUT.
      DIFFV=TOTVIN-TOTVOT
C
C7D-----GET PERCENT DIFFERENCE BETWEEN VOLUME IN AND VOLUME OUT.
      PDIFFV=100.*DIFFV/((TOTVIN+TOTVOT)/2)
C
C8-----PRINT DIFFERENCES AND PERCENT DIFFERENCES BETWEEN INPUT
C8-----AND OUTPUT RATES AND VOLUMES.
      WRITE(IOUT,299) DIFFV,DIFFR
      WRITE(IOUT,300) PDIFFV,PDIFFR
C
C9-----RETURN
      RETURN
C
C   ---FORMATS
C
260 FORMAT(1H0,///30X,'VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF'
1,' TIME STEP',I3,' IN STRESS PERIOD',I3/30X,77('-'))
265 FORMAT(1H0,19X,'CUMULATIVE VOLUMES',6X,'L**3',37X
1,' RATES FOR THIS TIME STEP',6X,'L**3/T'/20X,18('-'),47X,24('-')
2//26X,'IN:',68X,'IN:'/26X,'---',68X,'---')
275 FORMAT(1X,18X,4A4,' =',G14.5,39X,4A4,' =',G14.5)
286 FORMAT(1H0,26X,'TOTAL IN =',G14.5,47X,'TOTAL IN ='
1,G14.5)
287 FORMAT(1H0,24X,'OUT:',67X,'OUT:'/25X,4('-'),67X,4('-'))
298 FORMAT(1H0,25X,'TOTAL OUT =',G14.5,46X,'TOTAL OUT ='
1,G14.5)
299 FORMAT(1H0,26X,'IN - OUT =',G14.5,47X,'IN - OUT =',G14.5)
300 FORMAT(1H0,15X,'PERCENT DISCREPANCY =',F20.2
1,30X,'PERCENT DISCREPANCY =',F20.2,///)
C
      END

```

List of Variables for Module SBAS1V

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
DIFFR	Module	Sum of all inflow rates minus sum of all outflow rates (TOTRIN-TOTROT).
DIFFV	Module	Sum of all inflow volumes minus sum of all outflow volumes (TOTVIN-TOTVOT).
I	Module	Index.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for individual flows.
MSUM	Global	Counter for the budget entries and labels in VBVL and VBNM.
MSUM1	Module	MSUM-1.
PDIFFR	Module	Percent difference between the rate in and rate out.
PDIFFV	Module	Percent difference between the volume in and volume out.
TOTRIN	Module	Accumulator for the total of all inflow rates.
TOTROT	Module	Accumulator for the total of all outflow rates.
TOTVIN	Module	Accumulator for the total of all inflow volumes.
TOTVOT	Module	Accumulator for the total of all outflow volumes.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N) Rate for the current time step into the flow field. (2,N) Rate for the 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.

## CHAPTER 5

### BLOCK-CENTERED FLOW PACKAGE

#### Conceptualization and Implementation

The Block-Centered Flow (BCF) Package computes the conductance components of the finite-difference equation which determines flow between adjacent cells. It also computes the terms that represent the rate of movement of water to and from storage. To make the required calculations, it is assumed that a node is located at the center of each model cell and thus the name Block-Centered Flow is given to the package.

In chapter 1, the equation of flow for each cell in the model was developed as

$$\begin{aligned} & CV_{i,j,k-1/2}h_{i,j,k-1} + CC_{i-1/2,j,k}h_{i-1,j,k} + CR_{i,j-1/2,k}h_{i,j-1,k} \\ & + (- CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\ & - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h_{i,j,k} + CR_{i,j+1/2,k}h_{i,j+1,k} \\ & + CC_{i+1/2,j,k}h_{i+1,j,k} + CV_{i,j,k+1/2}h_{i,j,k+1} = RHS_{i,j,k}. \end{aligned} \quad (29)$$

The CV, CR, and CC coefficients are conductances between nodes--sometimes called "branch conductances." The HCOF and RHS coefficients are composed of external source terms and storage terms. Besides calculating the conductances and storage terms, the BCF Package calculates flow-correction terms that are added to HCOF and RHS to compensate for excess vertical flow that the flow equation calculates when part of a lower aquifer becomes unsaturated. The discussion of how all these calculations are made is divided into seven sections: Basic Conductance Equations, Horizontal Conductance, Vertical Conductance, A Variation of the Vertical Conductance Formulation, Storage Terms, Limitations on Use, and Data Requirements.

## Basic Conductance Equations

Conductance is a combination of several parameters used in Darcy's law. Darcy's law defines one-dimensional flow in a prism of porous material (fig. 23) as

$$Q = KA(h_2 - h_1)/L \quad (30)$$

where

Q is the flow ( $L^3t^{-1}$ );

K is the hydraulic conductivity of the material in the direction of flow ( $Lt^{-1}$ );

A is the cross-sectional area perpendicular to the flow ( $L^2$ );

$h_2 - h_1$  is the head difference across the prism parallel to flow (L); and

L is the length of the flow path (L).

Conductance, C, is defined as

$$C = KA/L. \quad (31)$$

Therefore, Darcy's law can be written as

$$Q = C(h_2 - h_1). \quad (32)$$

Another form of the conductance definition for horizontal flow in a prism is

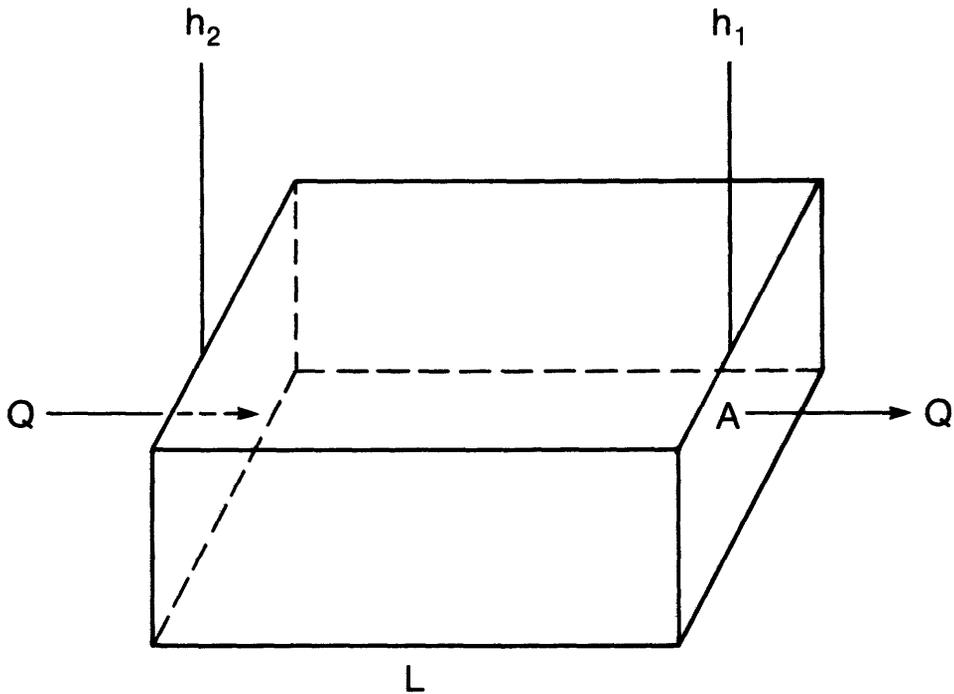
$$C = TW/L \quad (33)$$

where

T is transmissivity (K times thickness of the prism) in the direction of flow ( $L^2t^{-1}$ ); and

W is the width of the prism (L).

$$Q = \frac{KA (h_2 - h_1)}{L}$$



#### Explanation

- K Is Hydraulic Conductivity
- h<sub>2</sub> Is the Head at the Left End of the Prism
- h<sub>1</sub> Is the Head at the Right End of the Prism
- Q Is the Flow Rate from the Left End to the Right End
- L Is the Length of the Flow Path
- A Is the Cross Sectional Area Perpendicular to the Direction of Flow

Figure 23.—Prism of porous material to which Darcy's law may be applied.

Conductance is defined for a particular prism of material and for a particular direction. Thus for a prism of porous material, conductance in the three principal directions may be different.

If a prism of porous material consists of two or more subprisms in series and the conductance of each subprism is known, a conductance representing the entire prism can be calculated. The equivalent conductance is the rate of flow in the prism divided by the head change across the prism (fig. 24).

$$C = Q/(h_A - h_B). \quad (34)$$

Assuming continuity of head across each section in series gives the identity

$$\sum_{i=1}^n \Delta h_i = h_A - h_B. \quad (35)$$

Substituting for head change across each section using Darcy's law gives

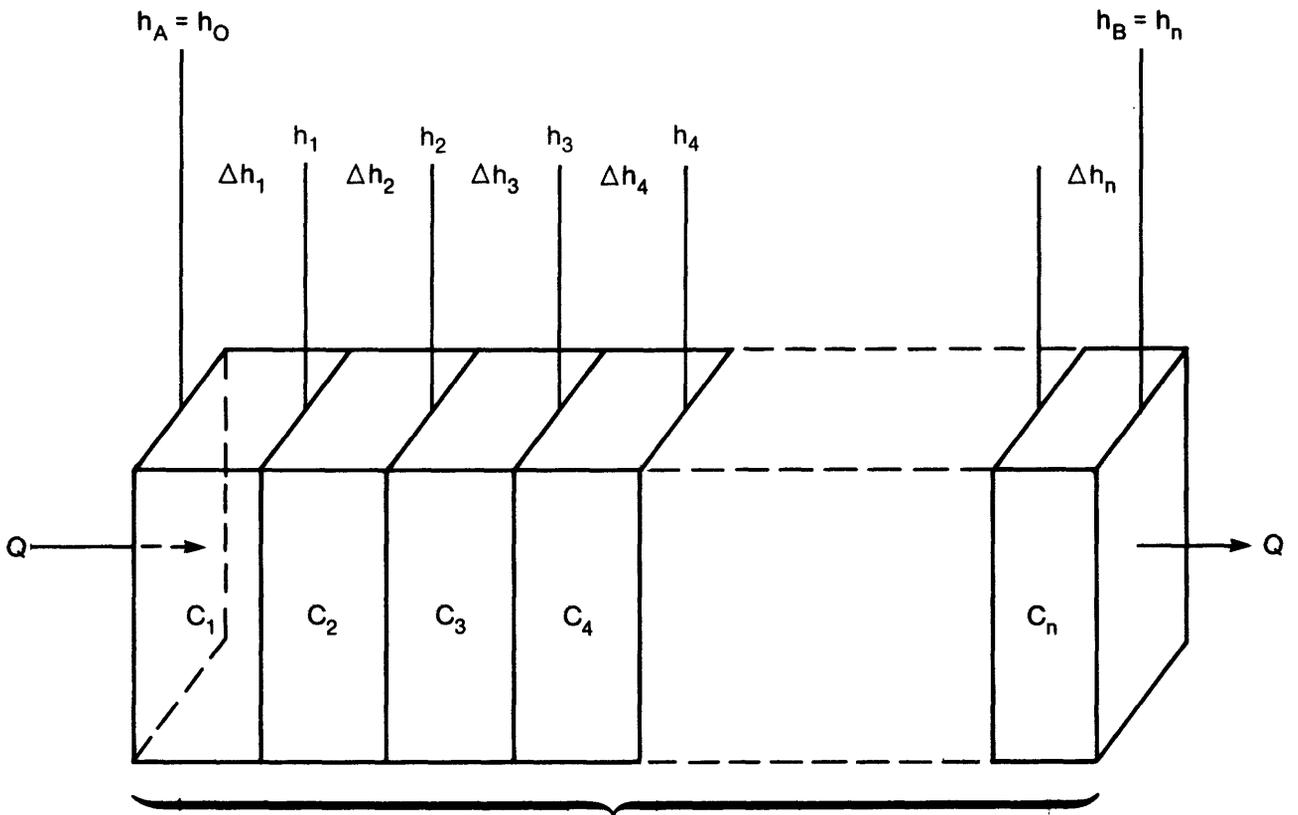
$$\sum_{i=1}^n \frac{q_i}{C_i} = h_A - h_B. \quad (36)$$

Since flow is one-dimensional and mass is conserved, all  $q_i$  are equal to total flow  $Q$ ; therefore,

$$Q \sum_{i=1}^n \frac{1}{C_i} = h_A - h_B \quad \text{and} \quad \frac{h_A - h_B}{Q} = \sum_{i=1}^n \frac{1}{C_i}. \quad (37)$$

By comparison with equation 34, it can be seen that

$$\frac{1}{C} = \sum_{i=1}^n \frac{1}{C_i}. \quad (38)$$



$$\frac{1}{C} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \dots + \frac{1}{C_n}$$

#### Explanation

$Q$  Is the Flow Rate

$C_m$  Is Conductance of Prism  $m$

$h_m$  Is Head at the Right Side of Prism  $m$

$\Delta h_m$  Is the Head Change Across Prism  $m$

$C$  Is the Conductance of the Entire Prism

Figure 24.—Calculation of conductance through several prisms in series.

When there are only two sections, the equivalent conductance reduces to

$$C = C_1 C_2 / (C_1 + C_2). \quad (39)$$

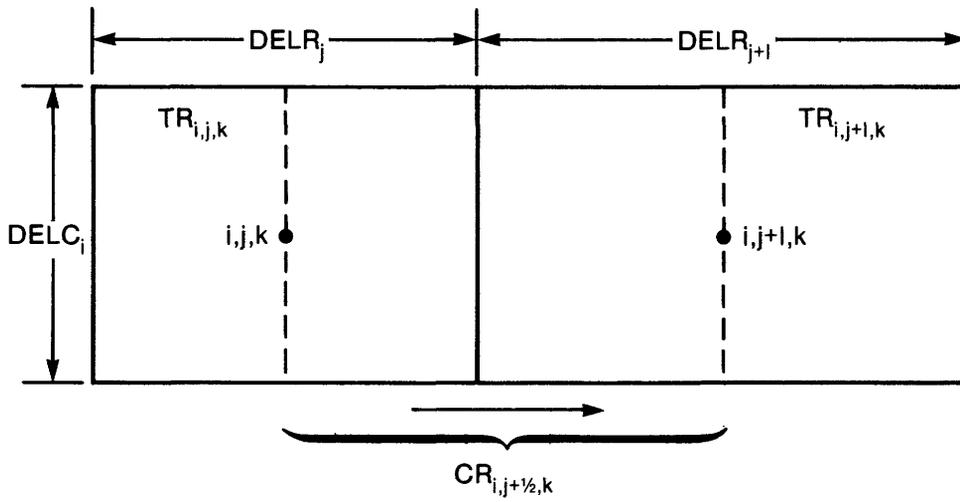
#### Horizontal Conductance

The finite-difference equations use the conductance between nodes of adjacent cells, "branch conductances," rather than simply the conductances within cells. Horizontal conductance terms CR and CC must be calculated between nodes that are adjacent horizontally. CR terms are oriented along rows and thus specify conductance between two nodes in the same row. Similarly, CC terms specify conductance between two nodes in the same column. To refer to conductance between nodes as opposed to conductance across cells, the subscript notation "1/2" is used. For example,  $CR_{i,j+1/2,k}$  represents the conductance between nodes  $i,j,k$  and  $i,j+1,k$ .

Figure 25 illustrates two cells along a row and the parameters used to calculate conductance between nodes in the cells. Two assumptions are made: (1) nodes are in the center of the cells and (2) transmissivity is uniform over a cell. Thus the conductance between the nodes is the equivalent conductance of two half cells in series ( $C_1$  and  $C_2$ ). Applying equation 39 gives

$$CR_{i,j+1/2,k} = C_1 C_2 / (C_1 + C_2). \quad (40)$$

Substituting in the conductance for each half cell by applying equation 33 gives



$$\frac{1}{CR_{i,j+1/2,k}} = \frac{1}{\frac{TR_{i,j,k} DELC_i}{DEL R_j}} + \frac{1}{\frac{TR_{i,j+1,k} DELC_i}{DEL R_{j+1}}}$$

$$CR_{i,j+1/2,k} = 2 DELC_i \times \frac{TR_{i,j,k} TR_{i,j+1,k}}{TR_{i,j,k} DEL R_{j+1} + TR_{i,j+1,k} DEL R_j}$$

#### Explanation

$TR_{i,j,k}$  Is Transmissivity in the Row Direction in Cell  $i,j,k$

$CR_{i,j+1/2,k}$  Is Conductance in the Row Direction Between Nodes  $i,j,k$  and  $i,j+1,k$

Figure 25.—Calculation of conductance between nodes using transmissivity and dimensions of cells.

$$CR_{i,j+1/2,k} = \frac{TR_{i,j,k} DELC_i}{1/2 DELR_j} + \frac{TR_{i,j+1,k} DELC_i}{1/2 DELR_{j+1}}$$

where

TR is transmissivity in the row direction ( $L^2t^{-1}$ );

DELR is the grid width along a row (L); and

DELC is the grid width along a column (L).

Simplification of this expression gives the final equation

$$CR_{i,j+1/2,k} = 2 DELC_i \frac{TR_{i,j,k} TR_{i,j+1,k}}{TR_{i,j,k} DELR_{j+1} + TR_{i,j+1,k} DELR_j} \quad (41)$$

The same process can be applied to the calculation of  $CC_{i+1/2j,k}$  giving

$$CC_{i+1/2,j,k} = 2 DELR_j \frac{TC_{i,j,k} TC_{i+1,j,k}}{TC_{i,j,k} DELC_{i+1} + TC_{i+1,j,k} DELC_i} \quad (42)$$

where

TC is the transmissivity in the column direction ( $L^2t^{-1}$ ).

Whenever transmissivity of both cells is zero, the conductance between the nodes in the cells is set equal to zero.

In a model layer which is confined, horizontal conductance will be constant for the simulation. If a layer is potentially unconfined, new values of horizontal conductance must be calculated as the head fluctuates. This is done at the start of each iteration. First, transmissivity is calculated from hydraulic conductivity and saturated thickness; then conductance is calculated from transmissivity and cell dimensions.

Transmissivity in a cell in the row direction is calculated using one of the following three equations

$$\begin{aligned} &\text{if } HNEW_{i,j,k} \geq TOP_{i,j,k}, \\ &\quad \text{then } TR_{i,j,k} = (TOP_{i,j,k} - BOT_{i,j,k}) HYR_{i,j,k}. \end{aligned} \quad (43)$$

$$\begin{aligned} &\text{if } TOP_{i,j,k} > HNEW_{i,j,k} > BOT_{i,j,k}, \\ &\quad \text{then } TR_{i,j,k} = (HNEW_{i,j,k} - BOT_{i,j,k}) HYR_{i,j,k} \end{aligned} \quad (44)$$

$$\begin{aligned} &\text{if } HNEW_{i,j,k} \leq BOT_{i,j,k}, \\ &\quad \text{then } TR_{i,j,k} = 0 \end{aligned} \quad (45)$$

where

$HYR_{i,j,k}$  is the hydraulic conductivity of cell  $i,j,k$  in the row direction ( $Lt^{-1}$ );

$TOP_{i,j,k}$  is the elevation of the top of cell  $i,j,k$  (L); and

$BOT_{i,j,k}$  is the elevation of the bottom of cell  $i,j,k$  (L).

Transmissivity in the column direction is the product of transmissivity in the row direction and a horizontal anisotropy factor specified by the user. The horizontal anisotropy factor is a constant for each layer. Conductances in each direction are calculated from transmissivity and cell dimensions. When head drops below the aquifer bottom (eq. 45), a cell is permanently set to no flow. There is no way to resaturate the cell again. This may cause errors in situations where reversals in water-level declines occur. Such reversals can occur as the result of a change in stress or as an error of the iterative-solution process. During iteration, heads may go lower than their final values at the end of a time step. This can cause a cell to erroneously change to no flow. The iterative solvers provide means to slow convergence in this situation.

## Vertical Conductance

Calculation of vertical conductance is conceptually similar to calculation of horizontal conductance. The finite-difference flow equation requires the conductance between two vertically adjacent nodes.  $CV_{i,j,k+1/2}$  is the conductance between nodes  $i,j,k$  and  $i,j,k+1$  in layers  $k$  and  $k+1$ . Applying equation 31 between two vertically adjacent model nodes (fig. 26) gives

$$CV_{i,j,k+1/2} = KV_{i,j,k+1/2} * DELR_j * DELC_i / DELV_{i,j,k+1/2} \quad (46)$$

where

$KV_{i,j,k+1/2}$  is the hydraulic conductivity between nodes  $i,j,k$  and  $i,j,k+1$  ( $Lt^{-1}$ ); and

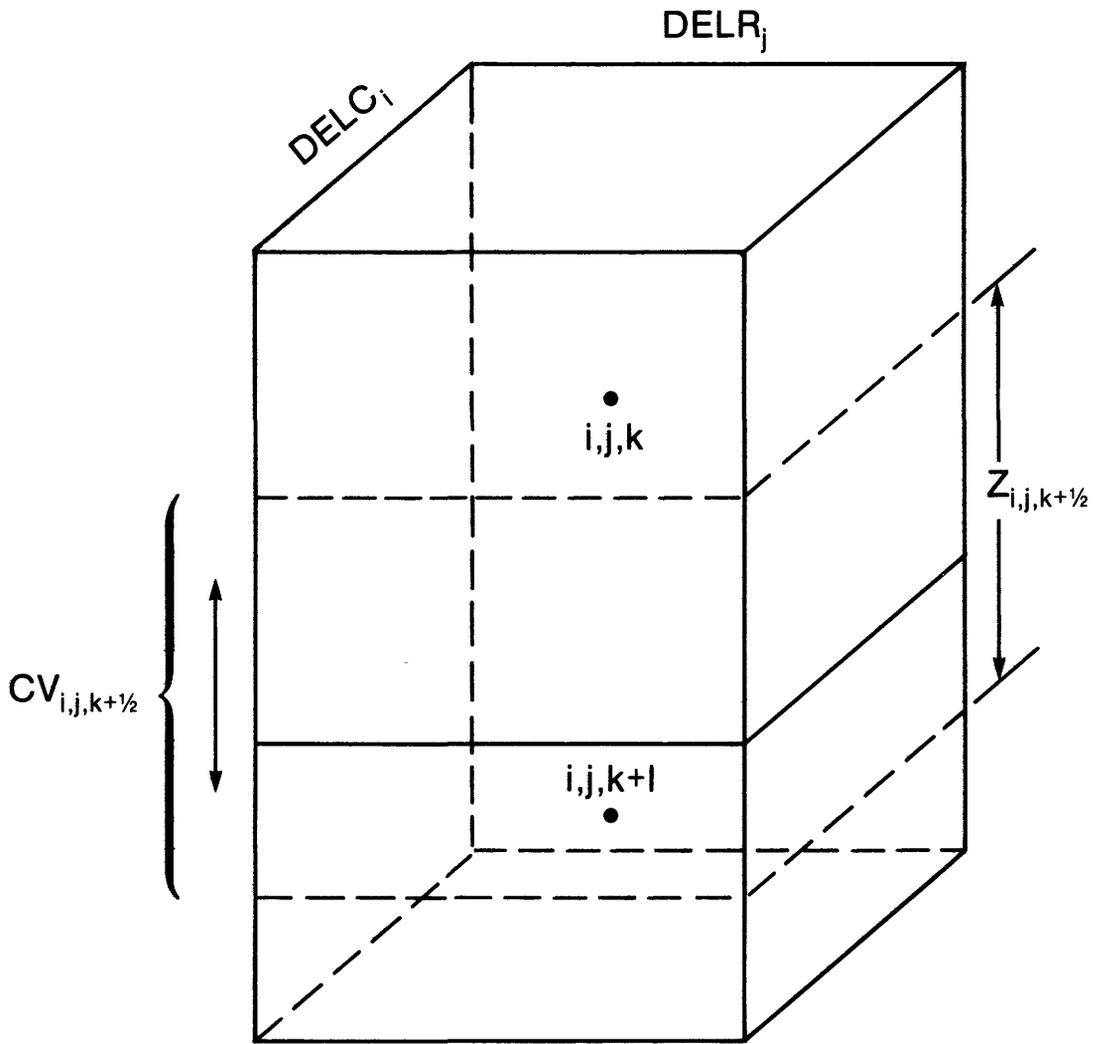
$DELV_{i,j,k+1/2}$  is the distance between nodes  $i,j,k$  and  $i,j,k+1$  (L).

Rather than specifying both vertical hydraulic conductivity and vertical grid spacing, a single term "Vcont" is specified. Vcont between nodes  $i,j,k$  and  $i,j,k+1$  is given by

$$Vcont_{i,j,k+1/2} = KV_{i,j,k+1/2} / DELV_{i,j,k+1/2}. \quad (47)$$

The program requires that Vcont between nodes be entered as input data rather than calculating it in the program.

Several methods can be used to calculate Vcont depending on the way that the aquifer system is discretized vertically. It is often desirable to use more than one method for calculating Vcont within the same simulation. Many of the methods of calculation could have been included in the model program, but the complexity of specifying where the various methods were to be applied and keeping track of data requirements would make the program



$$V \text{ cont}_{i,j,k+1/2} = KV_{i,j,k+1/2} / Z_{i,j,k+1/2}$$

$$CV_{i,j,k+1/2} = V \text{ cont}_{i,j,k+1/2} \text{ DELR}_j \text{ DELC}_i$$

**Explanation**

$KV_{i,j,k+1/2}$  Is Hydraulic Conductivity Between Node (i,j,k) and Node (i,j,k+1)

$Z_{i,j,k+1/2}$  Is the Distance Between Node (i,j,k) and Node (i,j,k+1)

Figure 26.—Calculating vertical conductance between adjacent cells.

and its use unnecessarily complex. The complexity is avoided by allowing the user to calculate  $V_{cont}$  outside of the program using any method desired.

The reason for variation in the method of vertical discretization is the desire to distort the grid in the vertical direction to minimize the number of model layers required to simulate an aquifer system. Such a distortion is illustrated in figure 27. Each distorted cell is simulated as if it were rectangular so that flow may be approximated by the standard finite-difference equation. Such distortion causes the vertical dimension to vary at each cell within a layer rather than being a constant. While the distortion introduces an error in the finite-difference approximation, the error is generally acceptably small. The distortion of a grid vertically is in contrast to horizontal discretization where a rectilinear grid is used even if the physical boundaries of the aquifer system are irregular. The grid can be made fine enough horizontally to adequately approximate an irregular boundary.

The simplest method for calculating  $V_{cont}$  is to directly use the definition of  $V_{cont}$  (eq. 47). This requires that the average or effective value of vertical hydraulic conductivity between nodes be known. In the case of a single aquifer broken into two or more model layers, the value of vertical hydraulic conductivity is frequently assumed to be the same in each layer so the value between nodes is that same constant.

If the vertical hydraulic conductivity is not the same in two adjacent layers, then the conductance of two half cells in series can be calculated using equations 31 and 39.

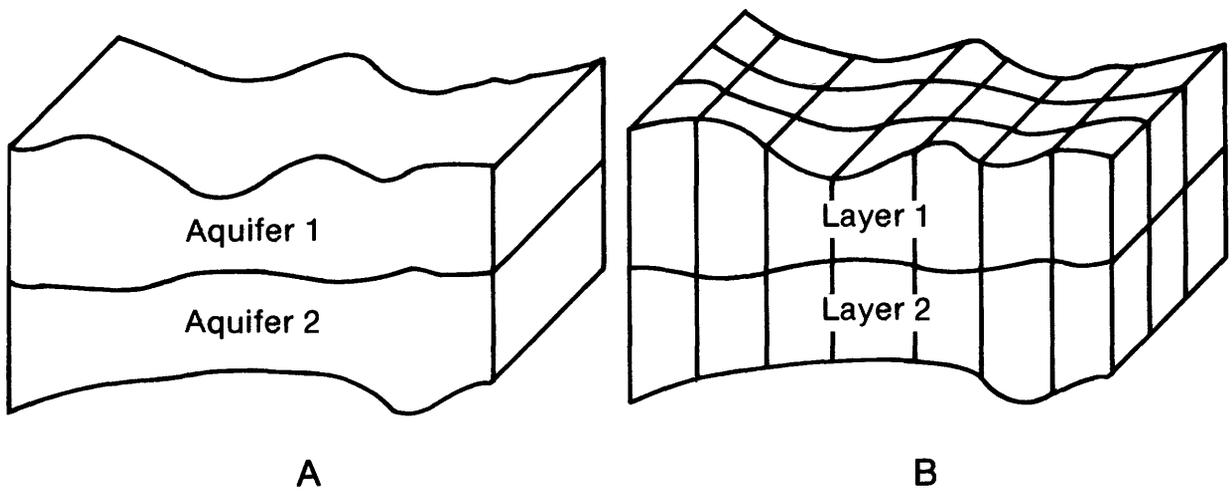


Figure 27.—Using a grid, distorted in the vertical direction, to represent lithologic units of varying thickness.

$$CV_{i,j,k+1/2} = \frac{\frac{KV_{i,j,k} DELR_j DELC_i}{1/2 DELV_{i,j,k}}}{\frac{KV_{i,j,k} DELR_j DELC_i}{1/2 DELV_{i,j,k}} + \frac{KV_{i,j,k+1} DELR_j DELC_i}{1/2 DELV_{i,j,k+1}}} \quad (48)$$

Equating the right hand side of equation 46 and the right hand side of equation 48 and rearranging yields

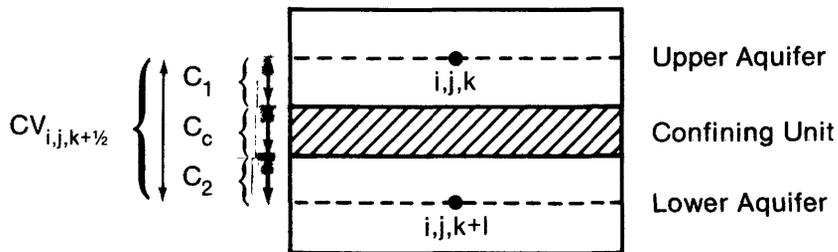
$$V_{cont_{i,j,k+1/2}} = \frac{KV_{i,j,k+1/2}}{DELV_{i,j,k+1/2}} = \frac{2}{\frac{DELV_{i,j,k}}{KV_{i,j,k}} + \frac{DELV_{i,j,k+1}}{KV_{i,j,k+1}}} \quad (49)$$

If the contrast in hydraulic conductivity is large between the two layers-- say,  $KV_{i,j,k}$  is much smaller than  $KV_{i,j,k+1}$ --then equation 49 can be approximated as

$$V_{cont} \approx 2KV_{i,j,k}/DELV_{i,j,k} \quad (50)$$

That is, the low  $V_{cont}$  of a confining bed may dominate the calculation so that  $V_{cont}$  of the aquifer can be ignored.

A third way to calculate  $V_{cont}$  comes from further simplification of vertical discretization. Figure 28 shows two aquifer layers separated by a confining bed. If storage in the confining bed and horizontal flow in the confining bed can be ignored, there is no need to have nodes within the confining bed. The storage condition will be met if the simulation is steady state or if the confining bed is thin. Horizontal flow in the confining bed can be ignored when the transmissivity of the bed is much lower than either aquifer layer. Using equation 38 to write the inverse



$$\frac{1}{CV_{i,j,k+1/2}} = \frac{1}{C_1} + \frac{1}{C_c} + \frac{1}{C_2}$$

#### Explanation

- $CV_{i,j,k+1/2}$  Is Conductance Between Node (i,j,k) and Node (i,j,k+1)
- $C_1$  Is Vertical Conductance of the Lower Half of Cell i,j,k
- $C_c$  Is Vertical Conductance of the Confining Unit
- $C_2$  Is Vertical Conductance of the Top Half of Cell i,j,k+1

Figure 28.—Calculating vertical conductance between two cells which are separated by a confining unit.

of the equivalent conductance between the two nodes (two aquifer blocks in series with the confining bed) gives

$$1/CV_{i,j,k+1/2} = 1/C_1 + 1/C_2 + 1/C_3. \quad (51)$$

The conductances  $C_1$  and  $C_3$  represent only that part of the aquifer between the aquifer nodes and the confining layer. If  $C_1$  and  $C_3$  are assumed to be much larger than  $C_2$ , the equation can be approximated as

$$1/CV_{i,j,k+1/2} \approx 1/C_2 \quad (52)$$

or

$$CV_{i,j,k+1/2} \approx C_2. \quad (53)$$

Thus the vertical hydraulic conductivity  $V_{cont}$  is simply the confining bed divided by its thickness.

#### A Variation of the Vertical Conductance Formulation

A continuity equation for cell  $i,j,k$  (eq. 25) is

$$\begin{aligned} & CR_{i,j-1/2,k}(h_{i,j-1,k}^m - h_{i,j,k}^m) + CR_{i,j+1/2,k}(h_{i,j+1,k}^m - h_{i,j,k}^m) \\ & + CC_{i-1/2,j,k}(h_{i-1,j,k}^m - h_{i,j,k}^m) + CC_{i+1/2,j,k}(h_{i+1,j,k}^m - h_{i,j,k}^m) \\ & + CV_{i,j,k-1/2}(h_{i,j,k-1}^m - h_{i,j,k}^m) + CV_{i,j,k+1/2}(h_{i,j,k+1}^m - h_{i,j,k}^m) \\ & + P_{i,j,k} h_{i,j,k}^m + Q_{i,j,k} = SS_{i,j,k}(\Delta r_j \Delta c_i \Delta v_k) \frac{(h_{i,j,k}^m - h_{i,j,k}^{m-1})}{t_m - t_{m-1}}. \end{aligned} \quad (54)$$

Flow through the lower face of cell  $i,j,k$ ,  $q_{i,j,k+1/2}$ , is

$$q_{i,j,k+1/2} = CV_{i,j,k+1/2}(h_{i,j,k+1}^m - h_{i,j,k}^m). \quad (55)$$

This equation was developed under the assumption that both the upper and lower cells are saturated. There are, however, situations where the lower cell may not be saturated. Such a situation is shown in figure 29. The upper and lower cells are separated by a confining unit. Pumping from the lower cell has drawn the head in that cell below the top of the cell--the lower cell is unconfined. Flow from the upper cell into the lower cell is no longer dependent on the head in the lower cell but rather on the head on the lower side of the confining unit. Because the top of the lower cell is unsaturated, the pressure is atmospheric; therefore, the head right below the confining unit is the elevation of the top of the lower cell. The equation for flow from the upper cell into the lower cell should be

$$q_{i,j,k+1/2} = CV_{i,j,k+1/2}(h_{i,j,k}^m - TOP_{i,j,k+1}). \quad (56)$$

The difference between the two formulations given in equations 55 and 56 is

$$q = CV_{i,j,k+1/2}(TOP_{i,j,k+1} - h_{i,j,k+1}^m). \quad (57)$$

To change the finite-difference equation to reflect the fact that the lower cell is unconfined, and that equation 55 overstates the amount of water flowing from cell  $i,j,k$  into cell  $i,j,k+1$ , the right side of equation 57 should be subtracted from the left side of equation 54. However, that would change the form of the finite-difference equation and make the coefficient matrix nonsymmetric. Replacing the head at cell  $i,j,k+1$ ,  $h_{i,j,k+1}^m$  in equation 57 with the head from the previous iteration  $h_{i,j,k+1}^{m-1}$  gives an approximation to  $q$ .

$$q = CV_{i,j,k+1/2}(TOP_{i,j,k+1} - h_{i,j,k+1}^{m-1}). \quad (58)$$

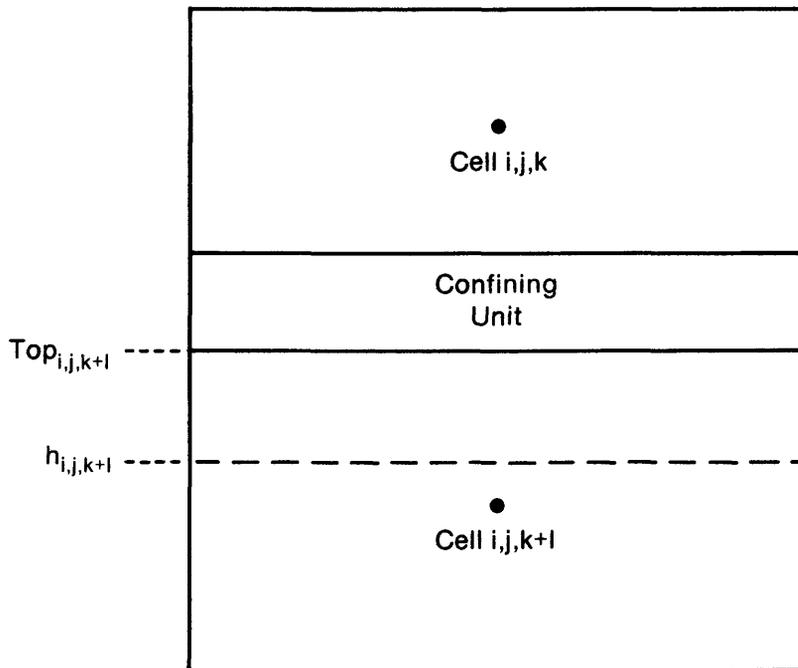


Figure 29.—Situation in which flow down into an unconfined cell may be overstated.

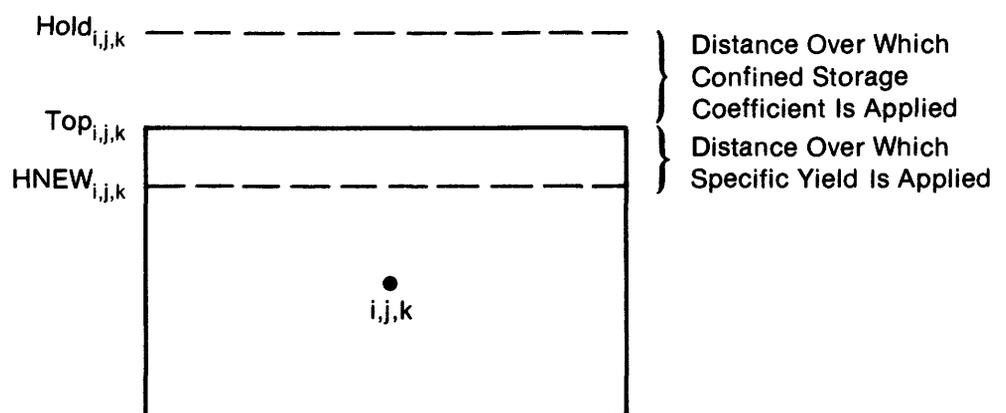


Figure 30.—A model cell which uses two storage factors during one iteration.

Rather than changing the form of the basic finite-difference equation, equation 58 is added to the right side of equation 54. In terms of equation 27, the term  $CV_{i,j,k+1/2}(TOP_{i,j,k+1} - h_{i,j,k+1}^{m-1})$  is added to the accumulator RHS.

A similar analysis can be made for the case where the cell  $i,j,k$  is unconfined and there is leakage from above. In that case, the amount by which the flow down into cell  $i,j,k$  is overstated is

$$q = CV_{i,j,k-1/2}(TOP_{i,j,k} - h_{i,j,k}). \quad (59)$$

The right side of equation 59 must be subtracted from the right side of equation 54. In terms of equation 27,  $CV_{i,j,k-1/2}(TOP_{i,j,k})$  is subtracted from the accumulator RHS;  $CV_{i,j,k-1/2}$  is subtracted from the accumulator HCOF.

#### Storage Terms

In the development of the finite-difference flow equation, the rate of change in storage of water in the cell was written as

$$Q = SA(HOLD_{i,j,k} - HNEW_{i,j,k})/DEL T \quad (60)$$

where

A is the area of the cell  $DEL R_j$  times  $DEL C_i$  ( $L^2$ );

DEL T is the length of the time step (t);

HOLD is the head at the end of the previous time step (L);

HNEW is the new head being calculated for the end of

the current time step (L); and

S is the dimensionless storage factor (specific yield or storage coefficient).

The BCF Package incorporates this component into RHS and HCOF. In the simple case of a strictly confined or unconfined aquifer, the implementation is straightforward. S is the specific yield for the unconfined case and storage coefficient for the confined case. The equation is broken into a constant part and a coefficient multiplied by HNEW for inclusion in RHS and HCOF. Equation 60 must be expanded to allow for the simulation of a model layer that can change from confined to unconfined and vice versa. Figure 30 illustrates the simulation when a cell converts from confined to unconfined. During a time step when a node changes from confined to unconfined, the storage equation is

$$Q = (ST1(HOLD-TOP)+ST2(TOP-HNEW))A/(DELTA) \quad (61)$$

where

TOP is the elevation of the top of the aquifer in the cell;

ST1 is the storage factor in effect at the start of the time step; and

ST2 is the current storage factor.

ST2 is equal to the storage coefficient if the head from the previous iteration is greater than the top of the layer (TOP); it is equal to specific yield if head from the previous iteration is less than TOP.

Equation 61 can be reorganized to show

$$Q = (-SNEW \times HNEW) + (SOLD(HOLD - TOP) + SNEW \times TOP) \quad (62)$$

where

SNEW is  $ST2 \times A/DELTA (L^2t^{-1})$ ; and

SOLD is  $ST1 \times A/DELTA (L^2t^{-1})$ .

The coefficient of the first term in equation 62 (-SNEW) is added to HCOF. The second term is subtracted from RHS.

## Limitations on Use

The approximations applied to the flow equation to simulate the effects of a water table (water-table transmissivity calculation, vertical leakage correction, and confined/unconfined storage conversion) were developed using the conceptualization of a layered aquifer system in which each aquifer is simulated by one model layer and these aquifer layers are separated by distinct confining units. If one attempts to use the water-table transmissivity calculation in the situation where several model layers are simulating the same aquifer and the water table is expected to traverse more than one layer, problems with cells incorrectly converting to no flow may occur. Because the conversion to no flow is irreversible, only declines in the water table can be simulated. Vertical conductance is left constant until a cell converts to no flow, and then is set to zero. This assumes there is a confining layer, which dominates vertical flow, below the model water-table layer. In particular, the model program may have difficulty handling a multilayer simulation of a single aquifer in which a well causes drawdown below the top model layer. The solver may attempt to convert cells to no-flow cells sooner than it should. This could cause the simulation to degenerate.

## Data Requirements

The formulations described here depend on the problem being simulated. The formulations are specified by assigning a numeric "layer-type code" to each model layer. The codes, which are stored in array LAYCON (layer configuration), are as follows:

Layer type = 0--The layer is strictly confined. Equation 60 with storage coefficient is used to calculate the rate of change in storage.

Transmissivity is constant throughout the simulation.

Input required for each cell in the layer:

Storage coefficient (only for transient simulations)

Transmissivity in the row direction

Vcont between the layer and the layer below

(if there is a layer below)

Layer type = 1--The layer is strictly unconfined. Equations 44 and 45 are used to calculate transmissivity each iteration. Equation 60 with specific yield is used to calculate rate of change in storage.

Input required for each cell in the layer:

Specific yield (only for transient simulations)

Hydraulic conductivity in the row direction

Aquifer bottom elevation

Vcont between this layer and the layer below

(if there is a layer below)

Layer type = 2--The layer is partially convertible between confined and unconfined. Transmissivity is constant throughout the simulation, but the confined/unconfined flow rate of change in storage (eq. 62) is used when appropriate, and vertical leakage from above is limited at unconfined cells. This is an approximation for a convertible layer, which is thick enough so that changes in transmissivity due to water-table fluctuations may be ignored.

Input required for each cell in the layer:

Storage coefficient (only for transient simulations)

Transmissivity in the row direction

Specific yield (only for transient simulations)

Elevation of the top of the aquifer

Vcont between this layer and the layer below

(if there is a layer below)

Layer type = 3--Fully convertible between confined and unconfined. Equation 62 is used as appropriate to calculate flow from storage. Equations 43-45 are used to calculate transmissivity every iteration. Vertical leakage from the aquifer above is limited at unconfined cells.

Input required for each cell in the layer:

Storage coefficient (only for transient simulations)

Hydraulic conductivity in the row direction

Elevation of the bottom of the aquifer

Specific yield (only for transient simulations)

Elevation of the top of the aquifer

Vcont between this layer and the layer below

(if there is a layer below)

Note that Vcont is included as part of the data to be entered for a layer. It represents, however, characteristics of two layers, the layer for which it is being read and the layer below. Thus, Vcont is not entered for the bottom layer. Vcont was included as part of layer data for programming convenience and ease of input.

Horizontal hydraulic conductance and transmissivity specified for the row direction are multiplied by a horizontal anisotropy factor (TRPY) to get hydraulic conductivity and transmissivity in the column direction. One horizontal anisotropy factor is specified by the user for each layer. Specific yield and storage coefficient are not needed for steady-state simulations. A flag (ISS) has been provided so that the user can specify that a simulation is steady state. When ISS is set, space is not allocated for specific yield or storage coefficient and storage calculations are skipped.

## Block-Centered Flow Package Input

Input for the Block-Centered Flow (BCF) Package is read from the unit specified in IUNIT(1).

FOR EACH SIMULATION

### BCF1AL

1. Data: ISS IBCFCB  
Format: I10 I10
2. Data: LAYCON(NLAY) (Maximum of 80 layers)  
Format: 40I2

(If there are 40 or fewer layers, use one record; otherwise, use two records.)

### BCF1RP

3. Data: TRPY(NLAY)  
Module: U1DREL
4. Data: DELR(NCOL)  
Module: U1DREL
5. Data: DELC(NROW)  
Module: U1DREL

A subset of the following two-dimensional arrays are used to describe each layer. The arrays needed for each layer depend on the layer type code (LAYCON) and whether the simulation is transient (ISS = 0) or steady state (ISS ≠ 0). If an array is not needed, it must be omitted. All of the arrays (items 6-12) for layer 1 are read first; then all of the arrays for layer 2, etc.

IF THE SIMULATION IS TRANSIENT

6. Data: sf1(NCOL,NROW)  
Module: U2DREL

IF THE LAYER TYPE CODE (LAYCON) IS ZERO OR TWO

7. Data: Tran(NCOL,NROW)  
Module: U2DREL

IF THE LAYER TYPE CODE (LAYCON) IS ONE OR THREE

8. Data: HY(NCOL,NROW)  
Module: U2DREL
9. Data: BOT(NCOL,NROW)  
Module: U2DREL

IF THIS IS NOT THE BOTTOM LAYER

10. Data: Vcont(NCOL,NROW)  
Module: U2DREL

IF THE SIMULATION IS TRANSIENT AND THE LAYER TYPE CODE (LAYCON) is TWO OR THREE

11. Data: sf2(NCOL,NROW)  
Module: U2DREL

IF THE LAYER TYPE CODE IS TWO OR THREE

12. Data: TOP(NCOL,NROW)  
Module: U2DREL

#### Explanation of Fields Used in Input Instructions

ISS--is the steady-state flag.

If ISS  $\neq$  0, the simulation is steady state.

If ISS = 0, the simulation is transient.

IBCFCB--is a flag and a unit number.

If IBCFCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If IBCFCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IBCFCB < 0, flow for each constant-head cell will be printed whenever ICBCFL is set.

LAYCON--is the layer type table. Each element holds the code for the respective layer. Read one value for each layer. There is a limit of 80 layers. Leave unused elements blank.

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

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

2 - confined/unconfined--Transmissivity of the layer is constant. The storage coefficient may be either confined or unconfined.

3 - confined/unconfined--Transmissivity of the layer varies. It is calculated from the saturated thickness and hydraulic conductivity. The storage coefficient may be either confined or unconfined. Vertical leakage from above is limited while unconfined.

TRPY--is a one-dimensional array containing an anisotropy factor for each layer. It is the ratio of transmissivity or hydraulic conductivity (whichever is being used) along a column to transmissivity or hydraulic conductivity along a row. Read one value per layer. Set to 1.0 for isotropic conditions. NOTE: This is one array with one value for each layer.

DELR--is the cell width along rows. Read one value for each of the NCOL columns.

DELC--is the cell width along columns. Read one value for each of the NROW rows.

sf1--is the primary storage factor. Read only for a transient simulation (steady-state flag, ISS, is 0). If the layer type (LAYCON) is type 1 (strictly unconfined), sf1 is equal to specific yield; otherwise, it is storage coefficient.

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

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

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

Vcont--is the vertical hydraulic conductivity divided by the thickness from a layer to the layer beneath it. Since there is not a layer beneath the bottom layer, Vcont cannot be specified for the bottom layer.

sf2--is the secondary storage factor. Read it only for layers where LAYCON is two or three and only if a transient simulation (steady-state flag, ISS, is zero). The secondary storage factor is always specific yield.

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



## Module Documentation for the Block-Centered Flow Package

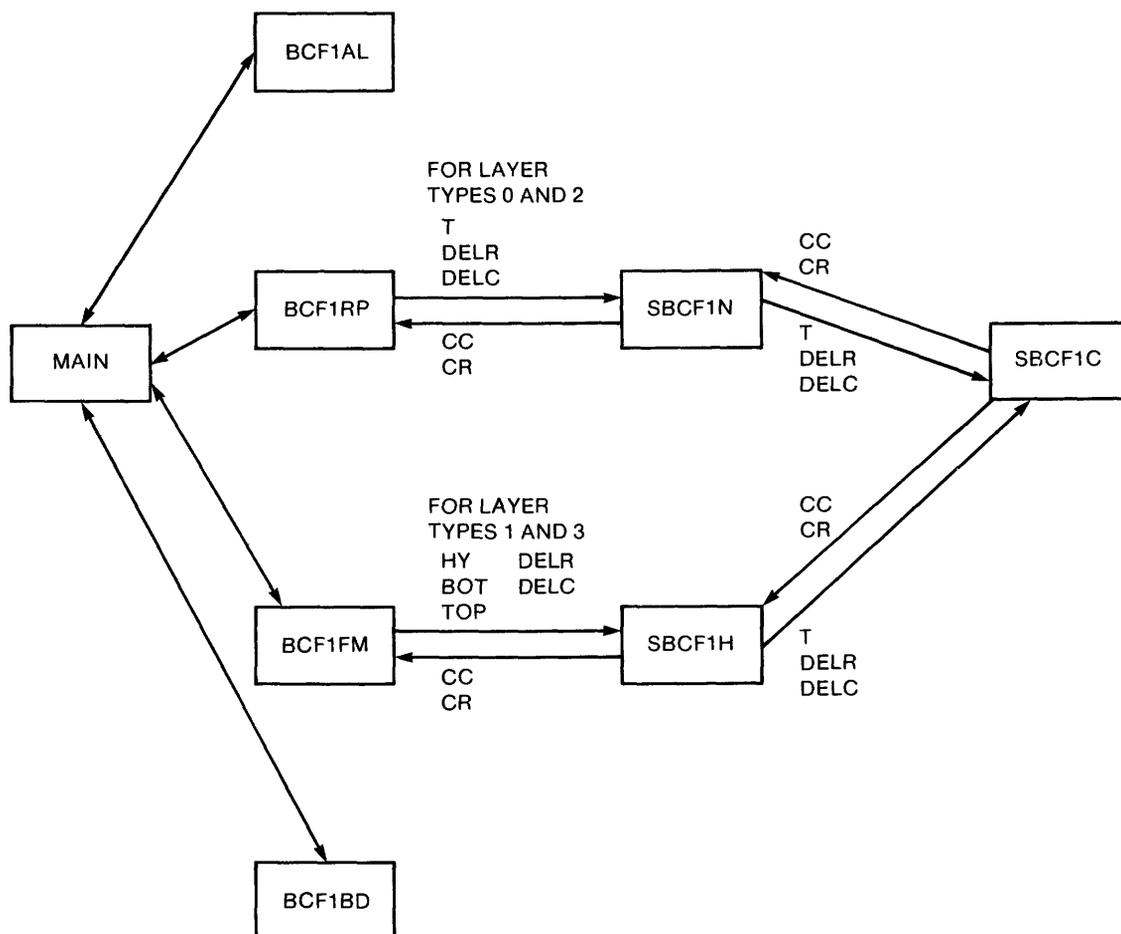
The Block-Centered Flow Package (BCF1) has four primary modules and three submodules. The relationship of the modules to MAIN and to each other is shown in figure 31. The flow of information used to calculate horizontal-hydraulic conductances (CC and CR) is shown for several of the modules. For example, BCF1RP passes transmissivity (T) and cell dimensions (DELR and DELC) to SBCF1N. Module SBCF1N then returns CC and CR to BCF1RP. The modules are:

### Primary Modules

- BCF1AL      Allocates space for data arrays.
- BCF1RP      Reads all data needed by the package, invokes SBCF1N to reconcile input transmissive values with the IBOUND array, and calculates storage capacities and constant conductances.
- BCF1FM      Calculates all coefficients of the system of equations that are not constant and invokes SBCF1H to calculate horizontal-branch conductances in partially saturated layers.
- BCF1BD      Calculates flow rates and accumulated flow volumes into and out of storage and constant-head boundaries. When cell-by-cell flow is specified, flow across all sides of each cell is also calculated.

### Submodules

- SBCF1N      Reconciles input transmissive values with the IBOUND array and calculates storage capacities and constant conductances. Invokes SBCF1C to calculate horizontal-branch conductances for layers where transmissivity is constant.
- SBCF1H      Calculates transmissivity for cells in layers where it depends on heads and invokes SBCF1C to calculate horizontal-branch conductances.
- SBCF1C      Calculates horizontal-branch conductance from cell transmissivity.
- SBCF1B      Calculates cell-by-cell flow terms across cell faces.
- SBCF1F      Calculates flow terms (both cell-by-cell and entries to overall budget) for flow to and from constant-head cells.



Explanation

CC	Conductance in the Column Direction	HY	Hydraulic Conductivity
CR	Conductance in the Row Direction	TOP	Elevation of the Top of a Layer
T	Transmissivity	BOT	Elevation of the Bottom of a Layer
DELR	Grid Spacing in the Row Direction	DELC	Grid Spacing in the Column Direction

Figure 31.—Relationship among the modules in the Block-Centered Flow Package.

## Narrative for Module BCF1AL

This module allocates space for data arrays for the Block-Centered Flow Package. It is done in the following order:

1. Print the message identifying the package.

2. Read and print the steady-state flag ISS and the cell-by-cell flow-term unit and flag (IBCFCB). Cell-by-cell flow terms for the BCF Package are flow to the right, flow forward, flow down, increase in storage, and flow to constant heads.

3. Read and print the layer-type code and count the number of layers which need the TOP array and the BOTTOM array.

- (a) Read the layer-type codes.

- 0 = confined

- 1 = unconfined

- 2 = confined/unconfined but transmissivity is constant

- 3 = confined/unconfined but transmissivity depends on head

- (b) Initialize the counters KT and KB in which the numbers of layers needing the TOP and BOTTOM are accumulated.

- (c) For each layer, print the layer-type code and determine if TOP and/or BOTTOM arrays are needed.

- (1) Print the layer number and the layer-type code.

- (2) If a layer other than the top layer is unconfined (type = 1), print an error message and STOP.

(3) If the layer type is one or three, add one to the BOTTOM counter, KB.

(4) If the layer type is two or three, add one to the TOP counter, KT.

4. Calculate the number of elements in the grid and in a layer.

5. Allocate space for the following arrays:

SC1     Primary storage factor;  
SC2     Secondary-storage factor (layer type 2 or 3 only);  
TRPY    Horizontal anisotropy factor;  
BOT     Bottom of layers (layer type 2 or 3 only);  
TOP     Top of layers (layer type 2 or 3 only); and  
HY      Hydraulic conductivity (layer type 1 or 3 only).

The following notes apply:

If the simulation is transient (ISS = 0), storage factors are needed.

The number of vertical conductance arrays is one less than the number of layers.

6. Print the amount of space used by the BCF Package.

7. RETURN.

## Flow Chart for Module BCF1AL

ISS is the steady-state flag. If it is set (ISS = 1), the simulation is steady state (storage is not considered).

IBCFCB is a flag and a unit number.

If IBCFCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If IBCFCB = 0, cell-by-cell flow terms will not be printed or recorded.

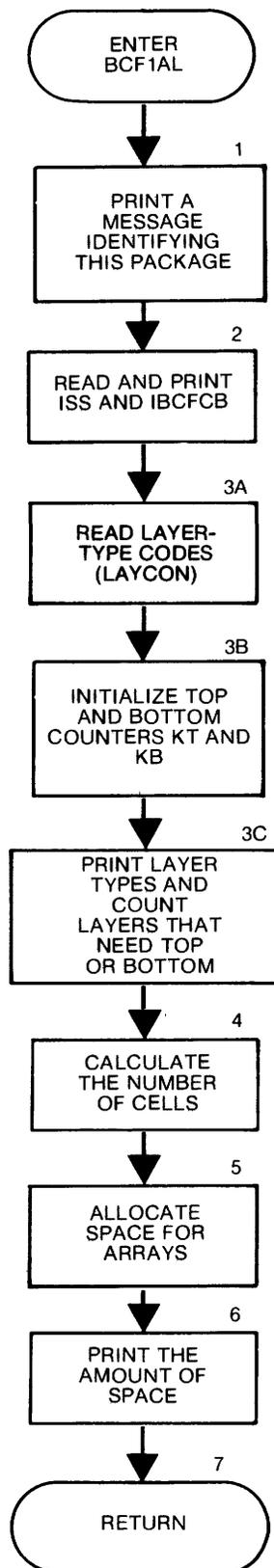
If IBCFCB < 0, flow from constant-head cells will be printed whenever ICBCFL is set.

LAYCON is a layer-type code (one for each layer).

- 0 - confined
- 1 - unconfined
- 2 - confined/unconfined but transmissivity is constant
- 3 - confined/unconfined

KT is a counter for the number of layers for which TOP is needed. (It is also the number of layers for which a secondary storage factor is needed.)

KB is a counter for the number of layers for which BOTTOM is needed. (It is also the number of layers for which hydraulic conductivity is needed.)



```

SUBROUTINE BCF1AL(ISUM,LENX,LCSC1,LCHY,LCBOT,
1      LCTOP,LCSC2,LCTRPY,IN,ISS,NCOL,NROW,NLAY,IOUT,IBCFCB)
C
C-----VERSION 0931 08DEC1983 BCF1AL
C
C      *****
C      ALLOCATE ARRAY STORAGE FOR BLOCK-CENTERED FLOW PACKAGE
C      *****
C
C      SPECIFICATIONS:
C      -----
C      COMMON /FLWCOM/LAYCON(80)
C      -----
C1-----IDENTIFY PACKAGE
      WRITE(IOUT,1)IN
      1 FORMAT(1H0,'BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1',
      1', 12/08/83', ' INPUT READ FROM UNIT',I3)
C
C2-----READ AND PRINT ISS (STEADY-STATE FLAG) AND IBCFCB (FLAG FOR
C2-----PRINTING OR UNIT# FOR RECORDING CELL-BY-CELL FLOW TERMS)
      READ(IN,2) ISS,IBCFCB
      2 FORMAT(2I10)
      IF(ISS.EQ.0) WRITE(IOUT,3)
      3 FORMAT(1X,'TRANSIENT SIMULATION')
      IF(ISS.NE.0) WRITE(IOUT,4)
      4 FORMAT(1X,'STEADY-STATE SIMULATION')
      IF(IBCFCB.GT.0) WRITE(IOUT,9) IBCFCB
      9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IBCFCB.LT.0) WRITE(IOUT,88)
      88 FORMAT(1X,'CONSTANT HEAD CELL-BY-CELL FLOWS WILL BE PRINTED')
C
C3-----READ TYPE CODE FOR EACH LAYER AND COUNT TOPS AND BOTTOMS
      IF(NLAY.LE.80) GO TO 50
      WRITE(IOUT,11)
      11 FORMAT(1H0,'YOU HAVE SPECIFIED MORE THAN 80 MODEL LAYERS'/1X,
      1 'SPACE IS RESERVED FOR A MAXIMUM OF 80 LAYERS IN ARRAY LAYCON')
      STOP
C
C3A-----READ LAYER TYPE CODES.
      50 READ(IN,51) (LAYCON(I),I=1,NLAY)
      51 FORMAT(40I2)
C      BOTTOM IS READ FOR TYPES 1,3    TOP IS READ FOR TYPES 2,3
      WRITE(IOUT,52)
      52 FORMAT(1X,5X,'LAYER  AQUIFER TYPE',/1X,5X,19('-'))
C
C3B-----INITIALIZE TOP AND BOTTOM COUNTERS.
      NBOT=0
      NTOP=0
C
C3C-----PRINT LAYER TYPE AND COUNT TOPS AND BOTTOMS NEEDED.
      DO 100 I=1,NLAY
C
C3C1----PRINT LAYER NUMBER AND LAYER TYPE CODE.

```

```

      L=LAYCON(I)
      WRITE(IOUT,7) I,L
      7 FORMAT(1X,I9,I10)
C
C3C2----ONLY THE TOP LAYER CAN BE UNCONFINED(LAYCON=1).
      IF(L.NE.1 .OR. I.EQ.1) GO TO 70
      WRITE(IOUT,8)
      8 FORMAT(1H0,'AQUIFER TYPE 1 IS ONLY ALLOWED IN TOP LAYER')
      STOP
C
C3C3----LAYER TYPES 1 AND 3 NEED A BOTTOM. ADD 1 TO KB.
      70 IF(L.EQ.1 .OR. L.EQ.3) NBOT=NBOT+1
C
C3C4----LAYER TYPES 2 AND 3 NEED A TOP. ADD 1 TO KT.
      IF(L.EQ.2 .OR. L.EQ.3) NTOP=NTOP+1
      100 CONTINUE
C
C
C
C4-----COMPUTE DIMENSIONS FOR ARRAYS.
      NRC=NROW*NCOL
      ISIZ=NRC*NLAY
C
C5-----ALLOCATE SPACE FOR ARRAYS. IF RUN IS TRANSIENT(ISS=0)
C5-----THEN SPACE MUST BE ALLOCATED FOR STORAGE.
      ISOLD=ISUM
      LCSC1=ISUM
      IF(ISS.EQ.0) ISUM=ISUM+ISIZ
      LCSC2=ISUM
      IF(ISS.EQ.0) ISUM=ISUM+NRC*NTOP
      LCTRPY=ISUM
      ISUM=ISUM+NLAY
      LCBOT=ISUM
      ISUM=ISUM+NRC*NBOT
      LCHY=ISUM
      ISUM=ISUM+NRC*NBOT
      LCTOP=ISUM
      ISUM=ISUM+NRC*NTOP
C
C6-----PRINT THE AMOUNT OF SPACE USED BY THE BCF PACKAGE.
      ISP=ISUM-ISOLD
      WRITE(IOUT,101) ISP
      101 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED BY BCF')
      ISUM1=ISUM-1
      WRITE(IOUT,102) ISUM1,LENX
      102 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,103)
      103 FORMAT(1X,' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C7-----RETURN
      RETURN
      END

```

List of Variables for Module BCF1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
I	Module	Index.
IBCFCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, flow from each constant-head cell will be printed whenever ICBCFL is set.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISIZ	Module	Number of cells in the grid.
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.
ISS	Package	Flag. = 0, simulation is transient. ≠ 0, simulation is steady state.
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	ISUM-1.
L	Module	Temporary storage for LAYCON(I).
LAYCON	Package	DIMENSION (80) Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
LCBOT	Package	Location in the X array of the first element of array BOT.
LCHY	Package	Location in the X array of the first element of array HY.
LCSC1	Package	Location in the X array of the first element of array SC1.
LCSC2	Package	Location in the X array of the first element of array SC2.
LCTOP	Package	Location in the X array of the first element of array TOP.
LCTRPY	Package	Location in the X array of the first element of array TRPY.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
NBOT	Module	Counter for the number of layers which need elevation of the bottom. Layers for which LAYCON = 1 or 3.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRC	Module	Number of cells in a layer.
NROW	Global	Number of rows in the grid.
NTOP	Module	Counter for the number of layers which need elevation of the top. LAYCON = 2 or 3.

## Narrative for Module BCF1RP

This module reads transmissivity along rows, hydraulic conductivity along rows, storage coefficients, vertical conductance, elevation of top of layer, and elevation of bottom of layer. It also calls SBCF1N to calculate parameters which are constant throughout simulation. It does this in the following order:

1. Call utility module UIDREL to read DELR, DELC, and TRPY which have one value for each column, row, and layer, respectively. TRPY is the ratio of transmissivity along columns to transmissivity along rows for each layer.

2. For each layer, use utility module UIDREL to read the properties of the porous medium. The data requirements for each layer are determined by the layer-type code.

(a) Find the address of the layer in the three-dimension arrays.

(b) If the simulation is transient ( $ISS = 0$ ), read the primary storage factor (storage coefficient if  $LAYCON = 0, 2,$  or  $3$ ; specific yield if  $LAYCON = 1$ ).

(c) For constant transmissivity layers ( $LAYCON = 0$  or  $2$ ), read the transmissivity.

(d) For variable transmissivity layers ( $LAYCON = 1$  or  $3$ ), read hydraulic conductivity and bottom.

(e) Read vertical-hydraulic conductivity divided by thickness. These values will be multiplied in the program by cell areas to get vertical conductance. Remember that for a layer, we need the conductance to the next lower layers. Therefore, we do not get a conductance for the lowest layer.

(f) If the simulation is transient and the layer type is two or three, read the secondary storage factor (specific storage).

(g) Read the top elevation if the layer type is two or three.

3. Call SBCF1N to calculate conductance and storage terms which are constant during the simulation and check to see that branch conductances agree with boundaries specified in the IBOUND array.

4. RETURN.

Flow Chart for Module BCF1RP

DELR is the grid spacing in the row direction.

DELC is the grid spacing in the column direction.

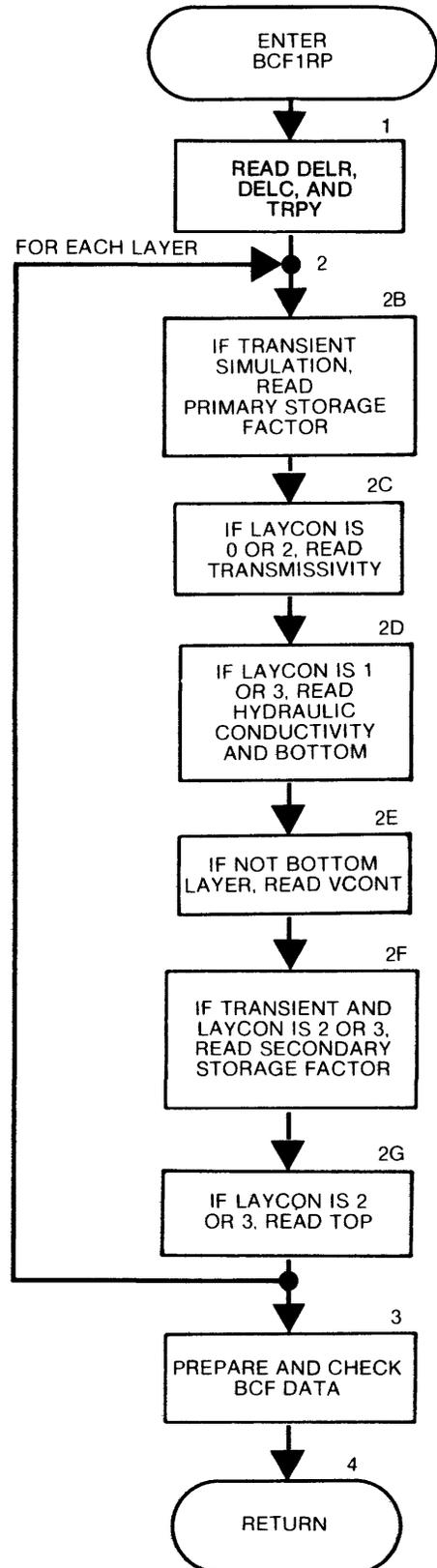
TRPY is the ratio of transmissivity in the column direction to transmissivity in the row direction.

LAYCON is a layer-type code (one for each layer).

- 0 - confined
- 1 - unconfined
- 2 - confined/unconfined but transmissivity is constant
- 3 - confined/unconfined

Primary Storage Factor is specific yield for unconfined layers (LAYCON = 1) and storage coefficient for confined or convertible layers (LAYCON = 0, 2, or 3).

Secondary Storage Factor is relevant only for convertible layers (LAYCON = 2 or 3); then it is specific yield.



```

SUBROUTINE BCF1RP(IBOUND,HNEW,SC1,HY,CR,CC,CV,DELR,DELCL,
1 BOT, TOP, SC2, TRPY, IN, ISS, NCOL, NROW, NLAY, NODES, IOUT)
C
C-----VERSION 1003 03MAY1983 BCF1RP
C
C*****
C READ AND INITIALIZE DATA FOR BLOCK-CENTERED FLOW PACKAGE
C*****
C
C SPECIFICATIONS:
C-----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NODES),SC1(NODES),HY(NODES),CR(NODES),CC(NODES),
1 CV(NODES), ANAME(6,10),DELR(NCOL),DELCL(NROW),BOT(NODES),
1 TOP(NODES),SC2(NODES),TRPY(NLAY),IBOUND(NODES)
C
C COMMON /FLWCOM/LAYCON(80)
C
C DATA ANAME(1,1), ANAME(2,1), ANAME(3,1), ANAME(4,1), ANAME(5,1),
1 ANAME(6,1) /4H ,4HPRIM,4HARY ,4HSTOR,4HAGE ,4HCOEF /
DATA ANAME(1,2), ANAME(2,2), ANAME(3,2), ANAME(4,2), ANAME(5,2),
1 ANAME(6,2) /4H ,4HTRAN,4HSMIS,4H AL,4HONG ,4HROWS /
DATA ANAME(1,3), ANAME(2,3), ANAME(3,3), ANAME(4,3), ANAME(5,3),
1 ANAME(6,3) /4H H,4HYD. ,4HCOND,4H AL,4HONG ,4HROWS /
DATA ANAME(1,4), ANAME(2,4), ANAME(3,4), ANAME(4,4), ANAME(5,4),
1 ANAME(6,4) /4HVERT,4H HYD,4H CON,4HD /T,4HHICK,4HNESS /
DATA ANAME(1,5), ANAME(2,5), ANAME(3,5), ANAME(4,5), ANAME(5,5),
1 ANAME(6,5) /4H ,4H ,4H ,4H ,4H BO,4HTTOM/
DATA ANAME(1,6), ANAME(2,6), ANAME(3,6), ANAME(4,6), ANAME(5,6),
1 ANAME(6,6) /4H ,4H ,4H ,4H ,4H ,4H TOP/
DATA ANAME(1,7), ANAME(2,7), ANAME(3,7), ANAME(4,7), ANAME(5,7),
1 ANAME(6,7) /4H SE,4HCOND,4HARY ,4HSTOR,4HAGE ,4HCOEF /
DATA ANAME(1,8), ANAME(2,8), ANAME(3,8), ANAME(4,8), ANAME(5,8),
1 ANAME(6,8) /4HCOLU,4HMN T,4HO RO,4HW AN,4HISOT,4HROPY/
DATA ANAME(1,9), ANAME(2,9), ANAME(3,9), ANAME(4,9), ANAME(5,9),
1 ANAME(6,9) /4H ,4H ,4H ,4H ,4H ,4HDELRL/
DATA ANAME(1,10), ANAME(2,10), ANAME(3,10), ANAME(4,10), ANAME(5,10),
1 ANAME(6,10) /4H ,4H ,4H ,4H ,4H ,4HDELCL/
C-----
C
C1-----CALCULATE NUMBER OF NODES IN A LAYER AND READ TRPY,DELR,DELCL
NIJ=NCOL*NROW
C
C CALL UIDREL(TRPY, ANAME(1,8), NLAY, IN, IOUT)
C CALL UIDREL(DELR, ANAME(1,9), NCOL, IN, IOUT)
C CALL UIDREL(DELCL, ANAME(1,10), NROW, IN, IOUT)
C
C2-----READ ALL PARAMETERS FOR EACH LAYER
KT=0
KB=0
DO 200 K=1, NLAY
C
C2A-----FIND ADDRESS OF EACH LAYER IN THREE DIMENSION ARRAYS.

```

```

IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) KB=KB+1
IF(LAYCON(K).EQ.2 .OR. LAYCON(K).EQ.3) KT=KT+1
LOC=1+(K-1)*NIJ
LOCB=1+(KB-1)*NIJ
LOCT=1+(KT-1)*NIJ
C
C2B-----READ PRIMARY STORAGE COEFFICIENT INTO ARRAY SC1 IF TRANSIENT
IF(ISS.EQ.0)CALL U2DREL(SC1(LOC),ANAME(1,1),NROW,NCOL,K,IN,IOUT)
C
C2C-----READ TRANSMISSIVITY INTO ARRAY CC IF LAYER TYPE IS 0 OR 2
IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) GO TO 100
CALL U2DREL(CC(LOC),ANAME(1,2),NROW,NCOL,K,IN,IOUT)
GO TO 110
C
C2D-----READ HYDRAULIC CONDUCTIVITY(HY) AND BOTTOM ELEVATION(BOT)
C2D-----IF LAYER TYPE IS 1 OR 3
100 CALL U2DREL(HY(LOCB),ANAME(1,3),NROW,NCOL,K,IN,IOUT)
CALL U2DREL(BOT(LOCB),ANAME(1,5),NROW,NCOL,K,IN,IOUT)
C
C2E-----READ VERTICAL HYCOND/THICK INTO ARRAY CV IF NOT BOTTOM LAYER
C2E----- READ AS HYCOND/THICKNESS -- CONVERTED TO CONDUCTANCE LATER
110 IF(K.EQ.NLAY) GO TO 120
CALL U2DREL(CV(LOC),ANAME(1,4),NROW,NCOL,K,IN,IOUT)
C
C2F-----READ SECONDARY STORAGE COEFFICIENT INTO ARRAY SC2 IF TRANSIENT
C2F-----AND LAYER TYPE IS 2 OR 3
120 IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 200
IF(ISS.EQ.0)CALL U2DREL(SC2(LOCT),ANAME(1,7),NROW,NCOL,K,IN,IOUT)
C
C2G-----READ TOP ELEVATION(TOP) IF LAYER TYPE IS 2 OR 3
CALL U2DREL(TOP(LOCT),ANAME(1,6),NROW,NCOL,K,IN,IOUT)
200 CONTINUE
C
C3-----PREPARE AND CHECK BCF DATA
CALL SBCF1N(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,ISS,
1          NCOL,NROW,NLAY,IOUT)
C
C4-----RETURN
RETURN
END

```

List of Variables for Module BCF1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ANAME	Module	Label for printout of input array.
BOT	Package	DIMENSION (NCOL,NROW,NBOT), Elevation of the bottom of each layer. (NBOT is the number of layers for which LAYCON = 1 or 3.)
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HY	Package	DIMENSION (NCOL,NROW,NBOT), Hydraulic conductivity of a cell. (NBOT is the number of layers where LAYCON = 1 or 3.)
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISS	Package	Flag. = 0, simulation is transient. ≠ 0, simulation is steady state.
K	Module	Index for layers.
KB	Module	Counter for the number of layers for which the bottom elevation is needed (LAYCON = 1 or 3).
KT	Module	Counter for the number of layers for which the top elevation is needed (LAYCON = 2 or 3).
LAYCON	Package	DIMENSION (80) Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
LOC	Module	Pointer to parts of the conductance arrays corresponding to particular layers.
LOCB	Module	Pointer to parts of the BOT and HY arrays corresponding to particular layers.

List of Variables for Module BCF1RP (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
LOCT	Module	Pointer to parts of the TOP and SC1 arrays corresponding to particular layers.
NCOL	Global	Number of columns in the grid.
NIJ	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.
SC1	Package	DIMENSION (NCOL,NROW,NLAY), Primary storage capacity of each cell (S*DELC*DELR).
SC2	Package	DIMENSION (NCOL,NROW,NTOP), Secondary storage capacity of each cell in the grid. (NTOP is the number of layers for which LAYCON = 2 or 3.)
TOP	Package	DIMENSION (NCOL,NROW,NTOP), Elevation of the top of the layers. (NTOP is the number of layers for which LAYCON = 2 or 3.)
TRPY	Package	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.

## Narrative for Module BCF1FM

This module calculates branch conductances which are not constant throughout the simulation, adds storage terms to the accumulators in which HCOF and RHS are formed, and adds terms to RHS and HCOF which correct for overestimation of flow down into partially saturated cells.

1. For each layer in which transmissivity varies with head (LAYCON = 1 or 3), call submodule SBCF1H to calculate branch conductance.

2. If the simulation is transient, calculate storage terms (STEPS 3-5) for each layer. If the simulation is steady state, GO TO STEP 6.

3. Determine if there is one storage factor or two.

4. If there is only one storage factor (LAYCON = 0 or 1), use it to calculate storage terms and add them to the right hand side (RHS) and the h-coefficient (HCOF).

5. If there are two storage factors, then, using head at the beginning of the time step (HOLD), determine the storage factor at the beginning of the time step (SOLD) and use the latest estimate of head at the end of the time step (HNEW) to determine the storage factor at the end of the time step (SNEW). Use SOLD and SNEW to calculate the storage terms to add to RHS and HCOF.

6. For each layer, determine if correction terms are needed for flow down into a partially saturated layer (STEPS 7-8).

7. If the layer is partially saturated and there is flow from above, calculate correction terms and add to RHS and HCOF.

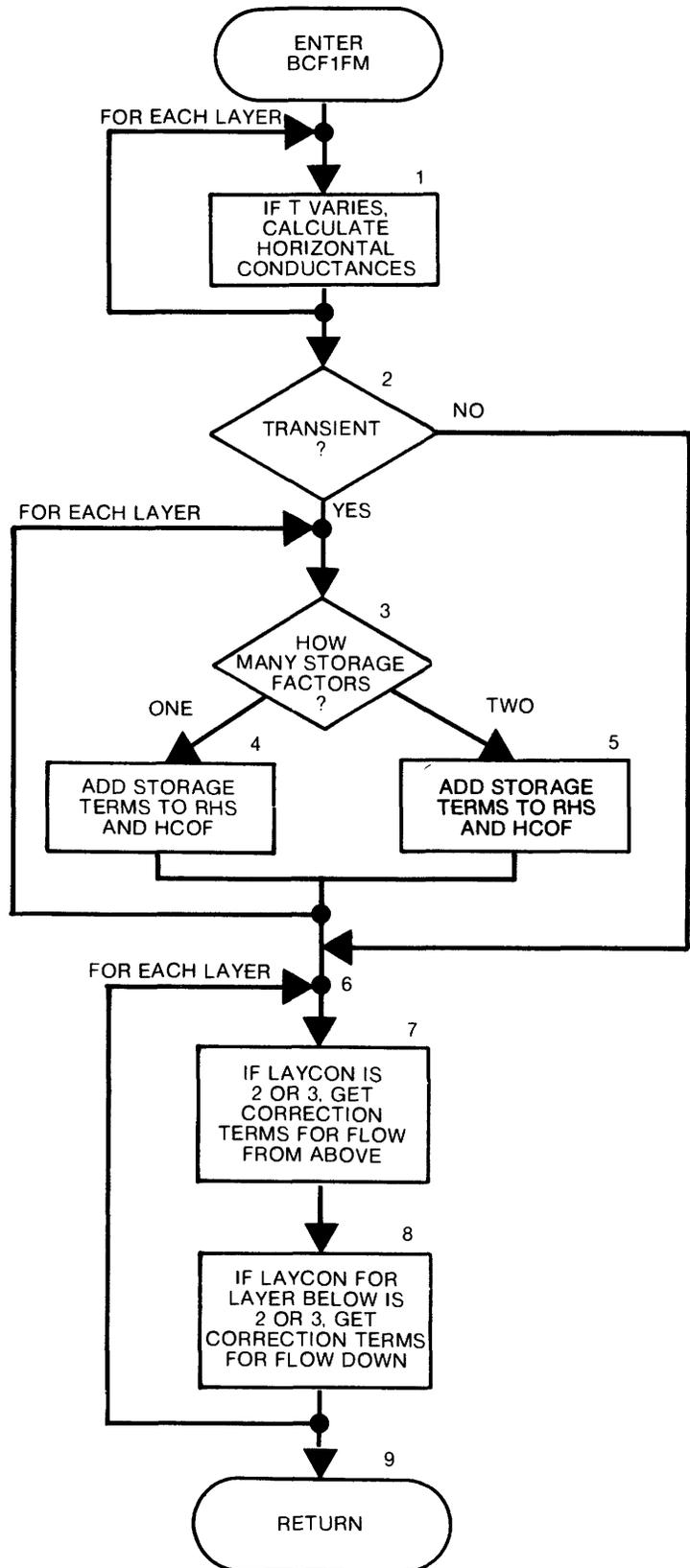
8. If this is not the bottom layer and the layer below is partially saturated, calculate the correction terms and add to RHS and HCOF.

9. RETURN.

## Flow Chart for Module BCF1FM

LAYCON is a layer-type code  
(one for each layer).

- 0 - confined
- 1 - unconfined
- 2 - confined/unconfined  
but transmissivity  
is constant
- 3 - confined/unconfined



```

SUBROUTINE BCF1FM(HCOF,RHS,HOLD,SC1,HNEW,IBOUND,CR,CC,CV,HY,TRPY,
1          BOT,TOP,SC2,DELR,DELC,DELT,ISS,KITER,KSTP,KPER,
2          NCOL,NROW,NLAY,IOUT)
C-----VERSION 1001 03MAY1983 BCF1FM
C
C *****
C ADD LEAKAGE CORRECTION AND STORAGE TO HCOF AND RHS, AND CALCULATE
C CONDUCTANCE AS REQUIRED
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY),
1 HOLD(NCOL,NROW,NLAY),SC1(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
2 IBOUND(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),
3 CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),
4 TRPY(NLAY),BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),DELR(NCOL),
5 DELC(NROW),SC2(NCOL,NROW,NLAY)
C
C COMMON /FLWCOM/LAYCON(80)
C -----
C KB=0
C KT=0
C
C1-----FOR EACH LAYER: IF T VARIES CALCULATE HORIZONTAL CONDUCTANCES
DO 100 K=1,NLAY
IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
C
C1A-----IF LAYER TYPE IS NOT 1 OR 3 THEN SKIP THIS LAYER.
IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.1) GO TO 100
KB=KB+1
C
C1B-----FOR LAYER TYPES 1 & 3 CALL SBCFHI TO CALCULATE
C1B-----HORIZONTAL CONDUCTANCES.
CALL SBCFHI(HNEW,IBOUND,CR,CC,CV,HY,TRPY,DELR,DELC,BOT,TOP,
1 K,KB,KT,KITER,KSTP,KPER,NCOL,NROW,NLAY,IOUT)
100 CONTINUE
C
C2-----IF THE SIMULATION IS TRANSIENT ADD STORAGE TO HCOF AND RHS
IF(ISS.NE.0) GO TO 201
TLED=1./DELT
KT=0
DO 200 K=1,NLAY
C
C3-----SEE IF THIS LAYER IS CONVERTIBLE OR NON-CONVERTIBLE.
IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) GO TO 150
C4-----NON-CONVERTIBLE LAYER, SO USE PRIMARY STORAGE
DO 140 I=1,NROW
DO 140 J=1,NCOL
RHO=SC1(J,I,K)*TLED
HCOF(J,I,K)=HCOF(J,I,K)-RHO
RHS(J,I,K)=RHS(J,I,K)-RHO*HOLD(J,I,K)
140 CONTINUE
GO TO 200
C
C5-----A CONVERTIBLE LAYER, SO CHECK OLD AND NEW HEADS TO DETERMINE
C5-----WHEN TO USE PRIMARY AND SECONDARY STORAGE
150 KT=KT+1
DO 180 I=1,NROW
DO 180 J=1,NCOL
C
C5A-----IF THE CELL IS EXTERNAL THEN SKIP IT.
IF(IBOUND(J,I,K).LE.0) GO TO 180
TP=TOP(J,I,KT)
RHO2=SC2(J,I,KT)*TLED
RHO1=SC1(J,I,K)*TLED

```

```

C
C5B-----FIND STORAGE FACTOR AT START OF TIME STEP.
      SOLD=RHO2
      IF(HOLD(J,I,K).GT.TP) SOLD=RHO1
C
C5C-----FIND STORAGE FACTOR AT END OF TIME STEP.
      HTMP=HNEW(J,I,K)
      SNEW=RHO2
      IF(HTMP.GT.TP) SNEW=RHO1
C
C5D-----ADD STORAGE TERMS TO RHS AND HCOF.
      HCOF(J,I,K)=HCOF(J,I,K)-SNEW
      RHS(J,I,K)=RHS(J,I,K) - SOLD*(HOLD(J,I,K)-TP) - SNEW*TP
C
      180 CONTINUE
C
      200 CONTINUE
C
C6-----FOR EACH LAYER DETERMINE IF CORRECTION TERMS ARE NEEDED FOR
C6-----FLOW DOWN INTO PARTIALLY SATURATED LAYERS.
      201 KT=0
      DO 300 K=1,NLAY
C
C7-----SEE IF CORRECTION IS NEEDED FOR LEAKAGE FROM ABOVE.
      IF(K.EQ.1) GO TO 250
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 250
      KT=KT+1
C
C7A-----FOR EACH CELL MAKE THE CORRECTION IF NEEDED.
      DO 220 I=1,NROW
      DO 220 J=1,NCOL
C
C7B-----IF THE CELL IS EXTERNAL(IBOUND<=0) THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 220
      HTMP=HNEW(J,I,K)
C
C7C-----IF HEAD IS ABOVE TOP THEN CORRECTION NOT NEEDED
      IF(HTMP.GE.TOP(J,I,KT)) GO TO 220
C
C7D-----WITH HEAD BELOW TOP ADD CORRECTION TERMS TO RHS AND HCOF.
      RHS(J,I,K)=RHS(J,I,K) + CV(J,I,K-1)*TOP(J,I,KT)
      HCOF(J,I,K)=HCOF(J,I,K) + CV(J,I,K-1)
      220 CONTINUE
C
C8-----SEE IF THIS LAYER MAY NEED CORRECTION FOR LEAKAGE TO BELOW.
      250 IF(K.EQ.NLAY) GO TO 300
      IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 300
      KTT=KT+1
C
C8A-----FOR EACH CELL MAKE THE CORRECTION IF NEEDED.
      DO 280 I=1,NROW
      DO 280 J=1,NCOL
C
C8B-----IF CELL IS EXTERNAL (IBOUND<=0) THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 280
C
C8C-----IF HEAD IN THE LOWER CELL IS LESS THAN TOP ADD CORRECTION
C8C-----TERM TO RHS.
      HTMP=HNEW(J,I,K+1)
      IF(HTMP.LT.TOP(J,I,KTT)) RHS(J,I,K)=RHS(J,I,K)
      1 - CV(J,I,K)*(TOP(J,I,KTT)-HTMP)
      280 CONTINUE
      300 CONTINUE
C
C9-----RETURN
      RETURN
      END

```

List of Variables for Module BCF1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BOT	Package	DIMENSION (NCOL,NROW,NBOT), Elevation of bottom of each layer. (NBOT is the number of layers for which LAYCON = 1 or 3.)
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
DELT	Global	Length of the current time step.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in cell (J,I,K) in the finite-difference equation.
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.
HTMP	Module	Temporary single precision HNEW(J,I,K).
HY	Package	DIMENSION (NCOL,NROW,NBOT), Hydraulic conductivity of a cell. (NBOT is the number of layers where LAYCON = 1 or 3.)
I	Module	Index for rows.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISS	Package	Flag. = 0, simulation is transient. ≠ 0, simulation is steady state.
J	Module	Index for columns.
K	Module	Index for layers.
KB	Module	Counter for layers for which bottom elevation is needed.
KITER	Global	Iteration counter. Reset at the start of each time step.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
KT	Module	Counter for layers for which top elevation is needed.
KTT	Module	Pointer to TOP array of layer immediately below layer K.

List of Variables for Module BCF1FM (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
LAYCON	Package	DIMENSION (80) Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RHO	Module	Storage coefficient for strictly confined or strictly unconfined layers.
RHO1	Module	Confined storage coefficient for convertible layers.
RHO2	Module	Unconfined storage coefficient for convertible layers.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of finite-difference equation. RHS is an accumulation of terms from several different packages.
SC1	Package	DIMENSION (NCOL,NROW,NLAY), Primary storage capacity of each cell (S*DELC*DELR).
SC2	Package	DIMENSION (NCOL,NROW,NTOP), Secondary storage capacity of each cell in the grid. (NTOP is the number of layers for which LAYCON = 2 or 3.)
SNEW	Module	Storage coefficient at the end of the time step for convertible layers.
SOLD	Module	Storage coefficient at the start of the time step for convertible layers.
TLED	Module	1/DELT.
TOP	Package	DIMENSION (NCOL,NROW,NTOP), Elevation of top of layers. (NTOP is the number of layers for which LAYCON = 2 or 3.)
TP	Module	Temporary variable for TOP(J,I,K).
TRPY	Package	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.

## Narrative for Module BCF1BD

Module BCF1BD calculates flow rates within the porous medium for use in the overall volumetric budget and calculates cell-by-cell flow terms for recording on disk. Flow rates to constant heads and from storage are accumulated and passed to the module BAS10T for inclusion in the budget. They are accumulated by sign so that flow into constant-head cells is separate from flow out of constant-head cells, and flow into storage is separate from flow out of storage. Flow rates to constant-head cells and from storage as well as flow across cell boundaries can be recorded on a cell-by-cell basis for use by other programs.

Flow from storage is calculated inside BCF1BD. Flow to constant-head cells and across cell boundaries is calculated in submodules SBCF1F and SBCF1B, respectively.

Module BCF1BD performs its tasks in the following order:

1. Clear the fields STOIN and STOUT in which flow out of and into storage, respectively, are accumulated.

2. If the user has specified that cell-by-cell flow terms should be recorded this time step (ICBCFL = 1) and has specified a unit number (IBCFCB) for cell-by-cell flow terms for the BCF Package, set the cell-by-cell flag (IBD).

3. If this is steady-state simulation, skip all of the calculations for flow from storage.

4. If cell-by-cell flow terms are to be saved (i.e., if IBD was set in STEP 2), clear the buffer (BUFF) in which they will be accumulated prior to printing.

5. For each cell in the grid, calculate flow from storage and move to accumulator (STEPS 6 AND 7).

6. Calculate flow from storage in the cell.

7. If the cell-by-cell rates are being recorded, store flow rate from storage in the buffer. Depending on the sign, add the flow from storage to the accumulators STOIN or STOUT.

8. If the cell-by-cell flag (IBD) is set, record the contents of the buffer.

9. Store the accumulated rates and volumes of flow from storage in table VBVL for inclusion in the overall volumetric budget. Store an appropriate label in the corresponding location in the table VBNM.

10. Call submodule SBCF1F to calculate flow from constant-head cells.

11. If the cell-by-cell flag (IBD) is set, call submodule SBCF1B to calculate and record the flow across cell boundaries.

12. RETURN.

## Flow Chart for Module BCF1BD

STOIN is an accumulator for flow terms having a positive sign (flow from storage into the flow system) for inclusion in the volumetric budget.

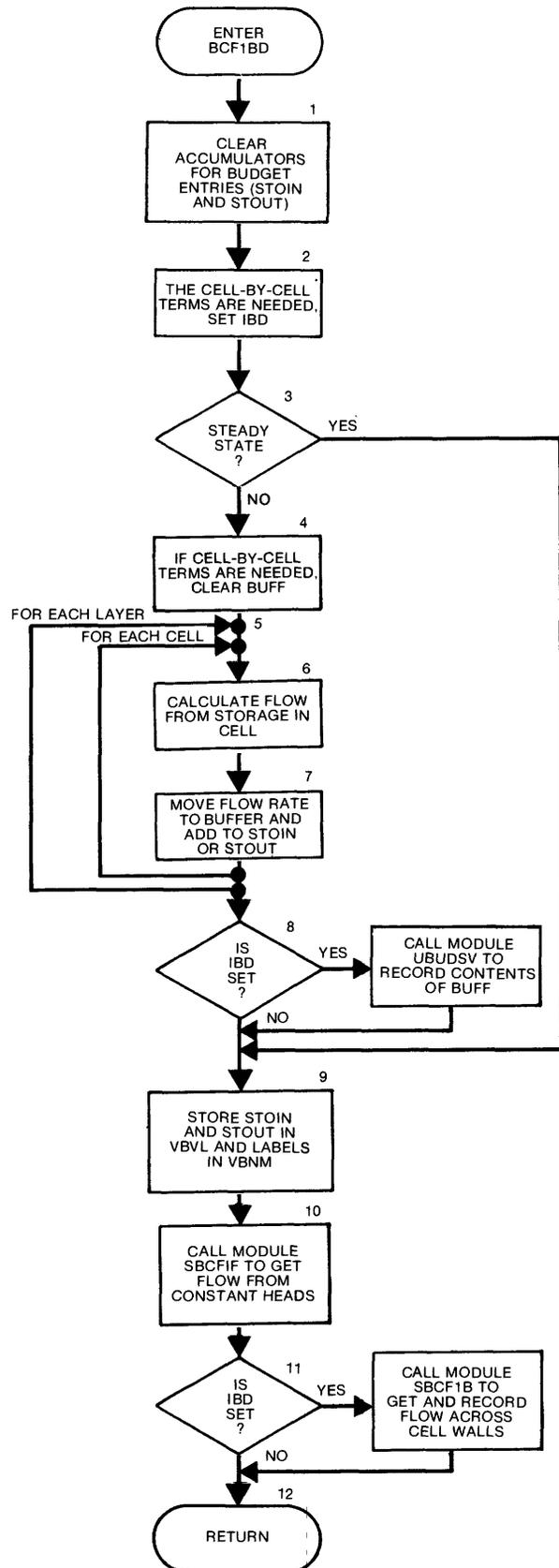
STOUT is an accumulator for flow terms having a negative sign (flow into storage and out of the flow system) for inclusion in the volumetric budget.

IBD is a flag which indicates that for this time step, BCF cell-by-cell flow terms should be recorded.

BUFF is a buffer where flow terms are gathered prior to recording them.

VBVL is a table of budget entries calculated by component-of-flow packages for use in calculating the volumetric budget.

VBNM is a table of labels for budget terms.



```

SUBROUTINE BCF1BD(VBNM,VBVL,MSUM,HNEW,IBOUND,HOLD,SC1,CR,CC,CV,
1  TOP,SC2,DELT,ISS,NCOL,NROW,NLAY,KSTP,KPER,IBCFCB,
2  ICBCFL,BUFF,IOUT)
C-----VERSION 1250 28DEC1983 BCF1BD
C
C *****
C COMPUTE BUDGET FLOW TERMS FOR BCF -- STORAGE, CONSTANT HEAD, AND
C FLOW ACROSS CELL WALLS
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
1  HOLD(NCOL,NROW,NLAY), SC1(NCOL,NROW,NLAY),
2  CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
3  CV(NCOL,NROW,NLAY), VBNM(4,20), VBVL(4,20),
4  SC2(NCOL,NROW,NLAY),
5  TOP(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
C
C COMMON /FLWCOM/LAYCON(80)
C
C DIMENSION TEXT(4)
C
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' ',' ' STO','RAGE'/
C -----
C1-----INITIALIZE BUDGET ACCUMULATORS
      STOIN=0.
      STOUT=0.
C
C2-----IF CELL-BY-CELL FLOWS ARE NEEDED THEN SET FLAG IBD.
      IBD=0
      IF(ICBCFL.NE.0 .AND. IBCFCB.GT.0) IBD=1
C
C3-----IF STEADY STATE THEN SKIP ALL STORAGE CALCULATIONS
      IF(ISS.NE.0) GO TO 305
C
C4-----IF CELL-BY-CELL FLOWS ARE NEEDED (IBD IS SET) CLEAR BUFFER
      IF(IBD.EQ.0) GO TO 220
      DO 210 K=1,NLAY
      DO 210 I=1,NROW
      DO 210 J=1,NCOL
      BUFF(J,I,K)=0.
210 CONTINUE
C
C5-----RUN THROUGH EVERY CELL IN THE GRID
220 KT=0
      DO 300 K=1,NLAY
      LC=LAYCON(K)
      IF(LC.EQ.3 .OR. LC.EQ.2) KT=KT+1
      DO 300 I=1,NROW
      DO 300 J=1,NCOL
C
C6-----CALCULATE FLOW FROM STORAGE (VARIABLE HEAD CELLS ONLY)
      IF(IBOUND(J,I,K).LE.0) GO TO 300

```

```

      HSING=HNEW(J,I,K)
C
C6A----CHECK LAYER TYPE TO SEE IF ONE STORAGE FACTOR OR TWO
      IF(LC.NE.3 .AND. LC.NE.2) GO TO 285
C
C6B----TWO STORAGE FACTORS
      TP=TOP(J,I,KT)
      SYA=SC2(J,I,KT)
      SCFA=SC1(J,I,K)
      SOLD=SYA
      IF(HOLD(J,I,K).GT.TP) SOLD=SCFA
      SNEW=SYA
      IF(HSING.GT.TP) SNEW=SCFA
      STRG=SOLD*(HOLD(J,I,K)-TP) + SNEW*TP - SNEW*HSING
      GO TO 288
C
C6C----ONE STORAGE FACTOR
      285 SC=SC1(J,I,K)
      STRG=SC*HOLD(J,I,K) - SC*HSING
C
C7-----STORE CELL-BY-CELL FLOW IN BUFFER AND ADD TO ACCUMULATORS
      288 IF(IBD.EQ.1) BUFF(J,I,K)=STRG/DELT
      IF(STRG) 292,300,294
      292 STOUT=STOUT-STRG
      GO TO 300
      294 STOIN=STOIN+STRG
C
      300 CONTINUE
C
C8-----IF IBD FLAG IS SET RECORD THE CONTENTS OF THE BUFFER
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,
      1 IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C9-----ADD TOTAL RATES AND VOLUMES TO VBVL & PUT TITLES IN VBNM
      305 VBVL(1,MSUM)=VBVL(1,MSUM)+STOIN
      VBVL(2,MSUM)=VBVL(2,MSUM)+STOUT
      VBVL(3,MSUM)=STOIN/DELT
      VBVL(4,MSUM)=STOUT/DELT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
      MSUM=MSUM+1
C
C10----CALCULATE FLOW FROM CONSTANT HEAD NODES
      CALL SBCF1F(VBNM,VBVL,MSUM,HNEW,IBOUND,CR,CC,CV,TOP,DELT,
      1 NCOL,NROW,NLAY,KSTP,KPER,IBD,IBCFCB,ICBCFL,BUFF,IOUT)
C
C11----CALCULATE AND SAVE FLOW ACROSS CELL BOUNDARIES IF C-B-C
C11----FLOW TERMS ARE REQUESTED.
      IF(IBD.NE.0) CALL SBCF1B(HNEW,IBOUND,CR,CC,CV,TOP,NCOL,NROW,NLAY,
      1 KSTP,KPER,IBCFCB,BUFF,IOUT)
C
C12----RETURN
      RETURN
      END

```

List of Variables for Module BCF1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains the conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
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.
HSING	Module	Temporary label for element of HNEW.
I	Module	Index for rows.
IBCFCB	Package	Flag and a unit number. > 0, unit number on which the cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, flow from each constant-head cell will be printed whenever ICBCFL is set.
IBD	Package	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
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 (flow to constant heads) will be either printed or recorded (depending on IBCFCB) for the current time step.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISS	Package	Flag. = 0, simulation is transient. ≠ 0, simulation is steady state.
J	Module	Index for columns.
K	Module	Index for layers.
KPER	Global	Stress period counter.

List of Variables for Module BCF1BD (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KSTP	Global	Time step counter. Reset at the start of each stress period.
KT	Module	Index for top of layers (also used for secondary storage terms).
LAYCON	Package	DIMENSION (80) Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
LC	Module	Temporary name for LAYCON(K).
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
SC	Module	Temporary name for the storage factor.
SCFA	Module	Temporary name for the primary storage factor.
SC1	Package	DIMENSION (NCOL,NROW,NLAY), Primary storage capacity of each cell (S*DELC*DELR).
SC2	Package	DIMENSION (NCOL,NROW,NTOP), Secondary storage capacity of each cell in the grid. (NTOP is the number of layers for which LAYCON = 2 or 3.)
SNEW	Module	Storage factor at the end of the time step.
SOLD	Module	Storage factor at the start of the time step.
STOIN	Module	Sum of decreases in storage from individual cells.
STOUT	Module	Sum of increases in storage for individual cells.
STRG	Module	Volume of flow into or out of storage in a single cell.
SYA	Module	Temporary name for the secondary storage factor.
TEXT	Module	Labels recorded along with the cell-by-cell flow terms.
TOP	Package	DIMENSION (NCOL,NROW,NTOP), Elevation of top of layers. (NTOP is the number of layers for which LAYCON = 2 or 3.)
TP	Module	Temporary label for TOP(J,I,K).
VBNM	Global	DIMENSION(4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION(4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the 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 SBCF1N

This module insures that the transmissive properties of each cell agree with the codes specified in the boundary array (IBOUND) and calculates (1) horizontal-branch conductance in layers where transmissivity is constant, (2) vertical-branch conductance, and (3) storage capacity.

The array IBOUND indicates the status of every cell in the grid with the following codes.

<u>Code</u>	<u>Status</u>
zero	inactive
positive	variable head
negative	constant head

The values in the IBOUND array are read by the BAS1RP module; transmissive properties are read by module BCF1RP. This module (SBCF1N) insures that all transmissive parameters are equal to zero for cells designated inactive by the IBOUND array and that cells are designated "inactive" if all transmissive parameters are equal to zero.

Module SBCF1N is called by module BCF1RP and calls submodule SBCF1C. The SBCF1N module performs these functions in the following order:

1. Check the cell to see if it is designated inactive (IBOUND = 0). If it is inactive, set the vertical leakance (temporarily stored in CV), transmissivity (temporarily stored in CC), and hydraulic conductivity equal to zero.

2. Check the cell that is designated active to insure that there is at least one nonzero transmissive parameter. If there are no such nonzero transmissive parameters, designate the cell inactive and print an error message.

- (a) If the transmissivity is constant (LAYCON = 0 or 2), the transmissivity or vertical-hydraulic conductivity must be nonzero.

- (b) If the transmissivity is a function of head (LAYCON = 1 or 3), the hydraulic conductivity or vertical conductance must be nonzero.

3. Calculate the horizontal-branch conductances for layers where the transmissivity is constant (LAYCON = 0 or 2). Submodule SBCF1C is invoked to calculate the branch conductance from the transmissivity and cell dimensions.

4. Multiply the vertical leakance between cells (temporarily stored in CV) by the cell dimensions to get the vertical conductance.

5. If the simulation is transient, multiply the primary storage factor by DELR and DELC to get the primary storage capacity (SC1).

6. If the layer is confined/unconfined, multiply the secondary storage factor by DELR and DELC to get the secondary storage capacity (SC2).

7. RETURN.

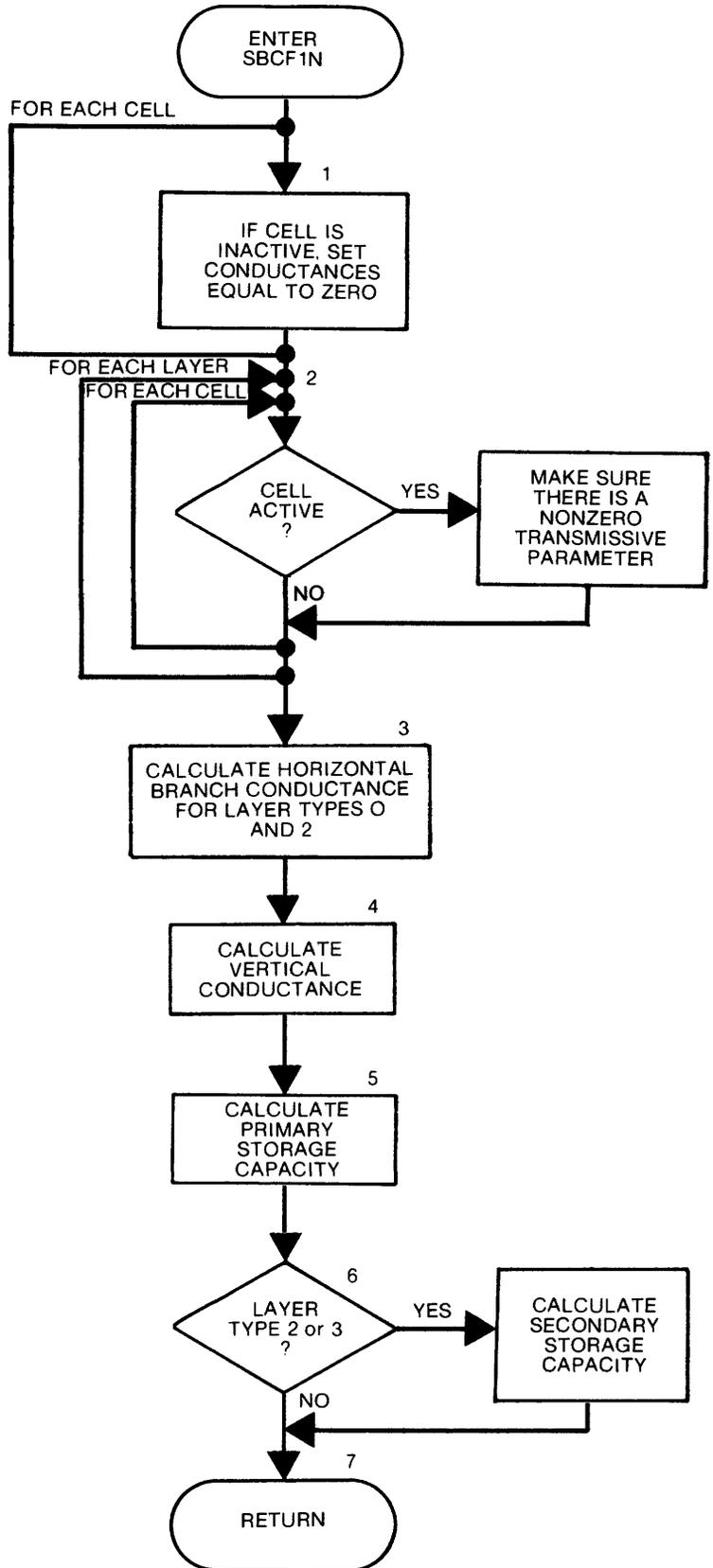
## Flow Chart for Module SBCF1N

LAYER TYPES are designated in the LAYCON table. Layer types are:

- 0 - confined
- 1 - unconfined
- 2 - constant/unconfined but transmissivity is constant
- 3 - confined/unconfined

Primary Storage Capacity is storage coefficient (specific yield for unconfined layers) times the cell area ( $DEL R * DEL C$ ).

Secondary Storage Capacity is specific yield (only for confined/unconfined layers) times the cell area.



```

          SUBROUTINE SBCF1N(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,
1          ISS,NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1007 03MAY1983 SBCF1N
C
C *****
C INITIALIZE AND CHECK BCF DATA
C *****
C
C SPECIFICATIONS:
C -----
C
C DOUBLE PRECISION HNEW,HCVN
C
C DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
1      ,SC1(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY)
2      ,CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY)
3      ,HY(NCOL,NROW,NLAY),TRPY(NLAY),DELR(NCOL),DELC(NROW)
4      ,SC2(NCOL,NROW,NLAY)
C
C COMMON /FLWCOM/LAYCON(80)
C -----
C1-----IF IBOUND=0, SET CV=0., CC=0., AND HY=0.
      KB=0
      DO 30 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) KB=KB+1
      DO 30 I=1,NROW
      DO 30 J=1,NCOL
      IF(IBOUND(J,I,K).NE.0) GO TO 30
      IF(K.NE.NLAY) CV(J,I,K)=0.
      IF(K.NE.1) CV(J,I,K-1)=0.
      CC(J,I,K)=0.
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) HY(J,I,KB)=0.
30 CONTINUE
C
C2-----INSURE THAT EACH ACTIVE CELL HAS AT LEAST ONE NON-ZERO
C2-----TRANSMISSIVE PARAMETER. IF NOT, CONVERT CELL TO NOFLOW.
      HCVN=888.88
      KB=0
      DO 60 K=1,NLAY
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) GO TO 55
C2A-----WHEN LAYER TYPE 0 OR 2, TRANSMISSIVITY OR CV MUST BE NONZERO
      DO 54 I=1,NROW
      DO 54 J=1,NCOL
      IF(IBOUND(J,I,K).EQ.0) GO TO 54
      IF(CC(J,I,K).NE.0.) GO TO 54
      IF(K.EQ.NLAY) GO TO 51
      IF(CV(J,I,K).NE.0.) GO TO 54
51 IF(K.EQ.1) GO TO 53
      IF(CV(J,I,K-1).NE.0.) GO TO 54
53 IBOUND(J,I,K)=0
      HNEW(J,I,K)=HCVN
      WRITE(IOUT,52) K,I,J
52 FORMAT(1X,'NODE (LAYER,ROW,COL)',3I4,
1      ' ELIMINATED BECAUSE ALL CONDUCTANCES TO NODE ARE 0')
54 CONTINUE
      GO TO 60

```

```

C
C2B-----WHEN LAYER TYPE IS 1 OR 3, HY OR CV MUST BE NONZERO
55 KB=KB+1
   DO 59 I=1,NROW
   DO 59 J=1,NCOL
   IF(IBOUND(J,I,K).EQ.0) GO TO 59
   IF(HY(J,I,KB).NE.0.) GO TO 59
   IF(K.EQ.NLAY) GO TO 56
   IF(CV(J,I,K).NE.0.) GO TO 59
56 IF(K.EQ.1) GO TO 57
   IF(CV(J,I,K-1).NE.0.) GO TO 59
57 IBOUND(J,I,K)=0
   HNEW(J,I,K)=HCNV
   CC(J,I,K)=0.
   WRITE(IOUT,52) K,I,J
59 CONTINUE
60 CONTINUE

C
C3-----CALCULATE HOR. CONDUCTANCE(CR AND CC) FOR CONSTANT T LAYERS
   DO 65 K=1,NLAY
   IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) GO TO 65
   CALL SBCFIC(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
65 CONTINUE

C
C4-----MULTIPLY VERTICAL LEAKANCE BY AREA TO MAKE CONDUCTANCE
   IF(NLAY.EQ.1) GO TO 69
   K1=NLAY-1
   DO 68 K=1,K1
   DO 68 I=1,NROW
   DO 68 J=1,NCOL
   CV(J,I,K)=CV(J,I,K)*DELR(J)*DELC(I)
68 CONTINUE

C
C5-----IF TRANSIENT MULTIPLY PRIMARY STORAGE FACTOR BY DELR &
C5-----DELC TO GET PRIMARY STORAGE CAPACITY(SC1).
69 IF(ISS.NE.0) GO TO 100
   KT=0
   DO 80 K=1,NLAY
   DO 70 I=1,NROW
   DO 70 J=1,NCOL
   SC1(J,I,K)=SC1(J,I,K)*DELR(J)*DELC(I)
70 CONTINUE

C
C6-----IF LAYER IS CONF/UNCONF MULTIPLY SECONDARY STORAGE FACTOR
C6-----BY DELR AND DELC TO GET SECONDARY STORAGE CAPACITY(SC2).
   IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 80
   KT=KT+1
   DO 75 I=1,NROW
   DO 75 J=1,NCOL
   SC2(J,I,KT)=SC2(J,I,KT)*DELR(J)*DELC(I)
75 CONTINUE

C
80 CONTINUE

C
C7-----RETURN
100 RETURN
END

```

List of Variables for Module SBCF1N

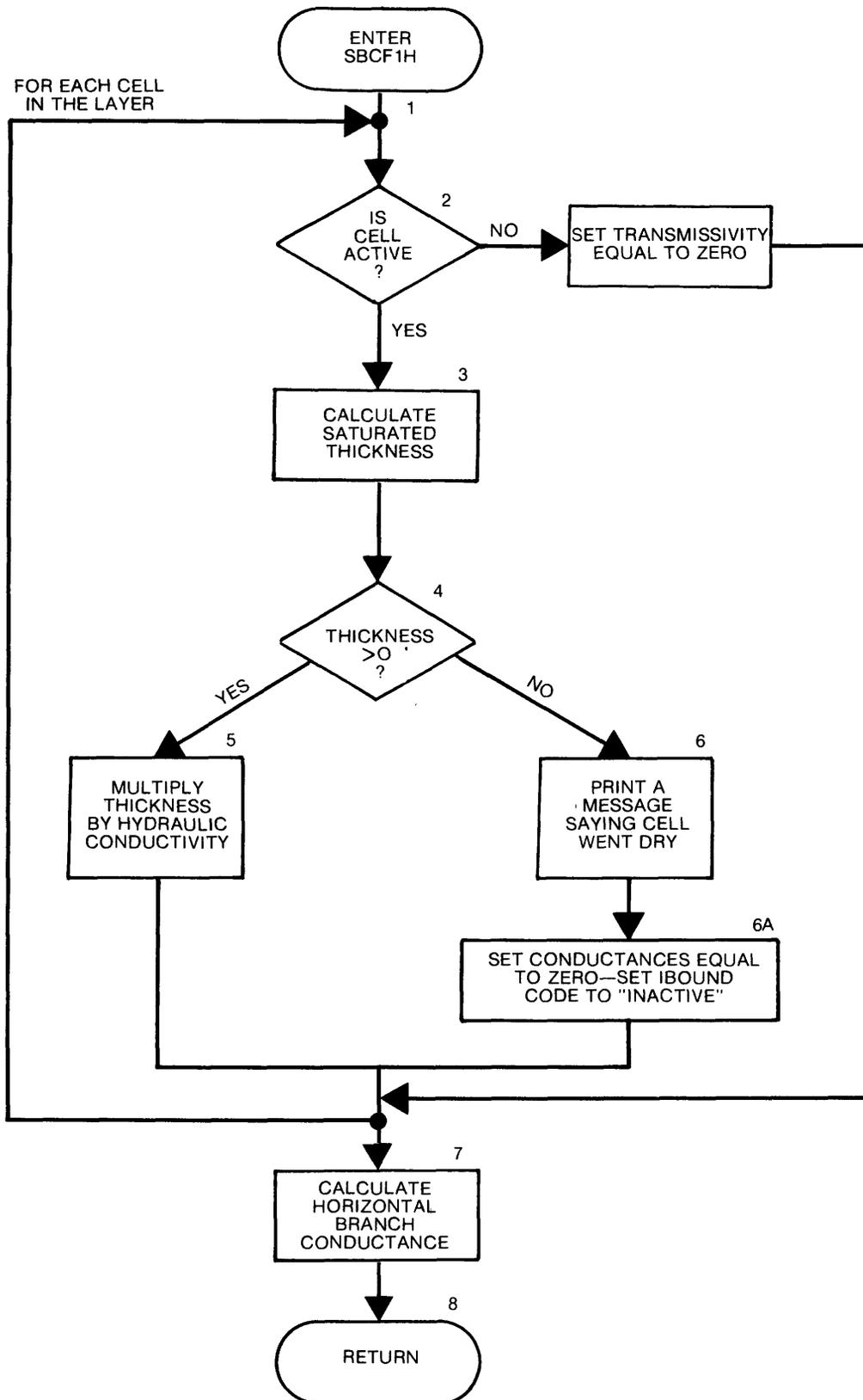
<u>Variable</u>	<u>Range</u>	<u>Definition</u>
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) AND (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
HCV	Module	Indicator in the HNEW array that the cell is inactive.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HY	Package	DIMENSION (NCOL,NROW,NBOT), Hydraulic conductivity of the cell. (NBOT is the number of layers where LAYCON = 1 or 3.)
I	Module	Index for rows.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISS	Package	Flag. = 0, simulation is transient. ≠ 0, simulation is steady state.
J	Module	Index for columns.
K	Module	Index for layers.
KB	Module	Index for bottom of layers.
KT	Module	Index for top of layers.
K1	Module	NLAY-1.
LAYCON	Package	DIMENSION(80), Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
NCOL	G	Number of columns in the grid.
NLAY	G	Number of layers in the grid.
NROW	G	Number of rows in the grid.
SC1	P	DIMENSION (NCOL,NROW,NLAY), Primary storage capacity of each cell (S*DELC*DELR).
SC2	P	DIMENSION (NCOL,NROW,NTOP), Secondary storage capacity of each cell in the grid. (NTOP is the number of layers for which LAYCON = 2 or 3.)
TRPY	P	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.

## Narrative for Module SBCF1H

Module SBCF1H calculates the horizontal-branch conductances (conductance between nodes) for a layer in which the transmissivity is a function of head (LAYCON = 1 or 3). It calculates the transmissivity internally and calls submodule SBCF1C to calculate the branch conductances. It is called by BCF1FM for each type 1 or type 3 layer at each iteration. Transmissivity is the product of hydraulic conductivity and saturated thickness. The saturated thickness of a completely saturated layer is computed as the elevation of the top (TOP) minus the elevation of the bottom (BOT), the thickness of the layer. For a partially saturated layer, saturated thickness is computed as the head in the cell minus the elevation of the bottom of the layer.

1. For each cell, calculate the transmissivity. DO STEPS 2-6.
2. If the cell is inactive, set the transmissivity equal to zero and move on to the next cell.
3. Calculate the thickness of the saturation. In a strictly unconfined layer, the thickness is the head (HNEW) minus the bottom (BOTTOM). In a confined/unconfined layer, the thickness is the head (HNEW) minus the bottom or the top (TOP) minus the bottom, whichever is greater.
4. Check to see if the saturated thickness is greater than zero.
5. If the thickness is greater than zero, the transmissivity of the cell is the thickness times the hydraulic conductivity.
6. If the saturated thickness is less than zero, the cell is dry. Print a message to that effect, set all branch conductances equal to zero, and set the boundary indicator (IBOUND) equal to zero.
7. Call submodule SBCF1C to calculate the horizontal-branch conductances for the layer.
8. RETURN.

Flow Chart for Module SBCF1H



```

SUBROUTINE SBCF1H(HNEW,IBOUND,CR,CC,CV,HY,TRPY,DELR,DEL
1,BOT,TOP,K,KB,KT,KITER,KSTP,KPER,NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1006 03MAY1983 SBCF1H
C
C *****
C COMPUTE CONDUCTANCE FROM SATURATED THICKNESS AND HYDRAULIC
C CONDUCTIVITY
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C 1, CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY), CV(NCOL,NROW,NLAY)
C 2, HY(NCOL,NROW,NLAY), TRPY(NLAY), DELR(NCOL), DELC(NROW)
C 3, BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY)
C
C COMMON /FLWCOM/LAYCON(80)
C -----
C
C1-----CALCULATE TRANSMISSIVITY AT EACH ACTIVE CELL. TRANSMISSIVITY
C1-----WILL BE STORED TEMPORARILY IN THE CC ARRAY.
C DO 200 I=1,NROW
C DO 200 J=1,NCOL
C
C2-----IF CELL IS INACTIVE THEN SET T=0 & MOVE ON TO NEXT CELL.
C IF (IBOUND(J,I,K).NE.0) GO TO 10
C CC(J,I,K)=0.
C GO TO 200
C
C3-----CALCULATE SATURATED THICKNESS.
C 10 HD=HNEW(J,I,K)
C IF (LAYCON(K).EQ.1) GO TO 50
C IF (HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
C 50 THCK=HD-BOT(J,I,KB)
C
C4-----CHECK TO SEE IF SATURATED THICKNESS IS GREATER THAN ZERO.
C IF (THCK.LE.0.) GO TO 100
C
C5-----IF SATURATED THICKNESS>0 THEN T=K*THICKNESS.
C CC(J,I,K)=THCK*HY(J,I,KB)
C GO TO 200
C
C6-----WHEN SATURATED THICKNESS < 0, PRINT A MESSAGE AND SET
C6-----TRANSMISSIVITY, IBOUND, AND VERTICAL CONDUCTANCE =0
C 100 WRITE(IOUT,150) J,I,K,KITER,KSTP,KPER
C 150 FORMAT(1H0,10('*'),'NODE',3I4,' (COL,ROW,LAYER) WENT DRY'
C 1 , ' AT ITERATION =',I3,' TIME STEP =',I3
C 2 , ' STRESS PERIOD =',I3)
C HNEW(J,I,K)=1.E30
C CC(J,I,K)=0.
C IBOUND(J,I,K)=0
C IF (K.LT.NLAY) CV(J,I,K)=0.
C IF (K.GT.1) CV(J,I,K-1)=0.
C GO TO 200
C 200 CONTINUE
C
C7-----COMPUTE HORIZONTAL BRANCH CONDUCTANCES FROM TRANSMISSIVITY
C CALL SBCF1C(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
C
C8-----RETURN
C RETURN
C END

```

List of Variables for Module SBCF1H

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BOT	Package	DIMENSION (NCOL,NROW,NBOT), Elevation of the bottom of each layer. (NBOT is the number of layers for which LAYCON = 1 or 3.)
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
HD	Module	Temporary label for an element in HNEW.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HY	Package	DIMENSION (NCOL,NROW,NBOT), Hydraulic conductivity of the cell. (NBOT is the number of layers where LAYCON = 1 or 3.)
I	Module	Index for rows.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
J	Module	Index for columns.
K	Module	Index for layers.
KB	Module	Index for bottom of layers.
KITER	Global	Iteration counter. Reset at the start of each time step.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
KT	Module	Index for tops of layers.
LAYCON	Package	DIMENSION(80), Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant) 3 - Layer confined/unconfined (transmissivity varies).
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
THCK	Module	Saturated thickness.
TOP	Package	DIMENSION (NCOL,NROW,NTOP), Elevation of top of layers. (NTOP is number of layers for which LAYCON = 2 or 3.)
TRPY	Package	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.

### Narrative for Module SBCF1C

The module SBCF1C calculates horizontal-branch conductances for a layer from transmissivity and cell dimensions. It is called by submodules SBCF1N and SBCF1H. Recall that the branch conductances between two nodes can be expressed by

$$C = C_1 C_2 / (C_1 + C_2).$$

However,  $C_1$  and  $C_2$  can be represented by

$$C_1 = T_1 W / (L_1 / 2)$$

$$C_2 = T_2 W / (L_2 / 2).$$

Thus,

$$C = 2T_1 T_2 W / (T_1 L_2 + T_2 L_1).$$

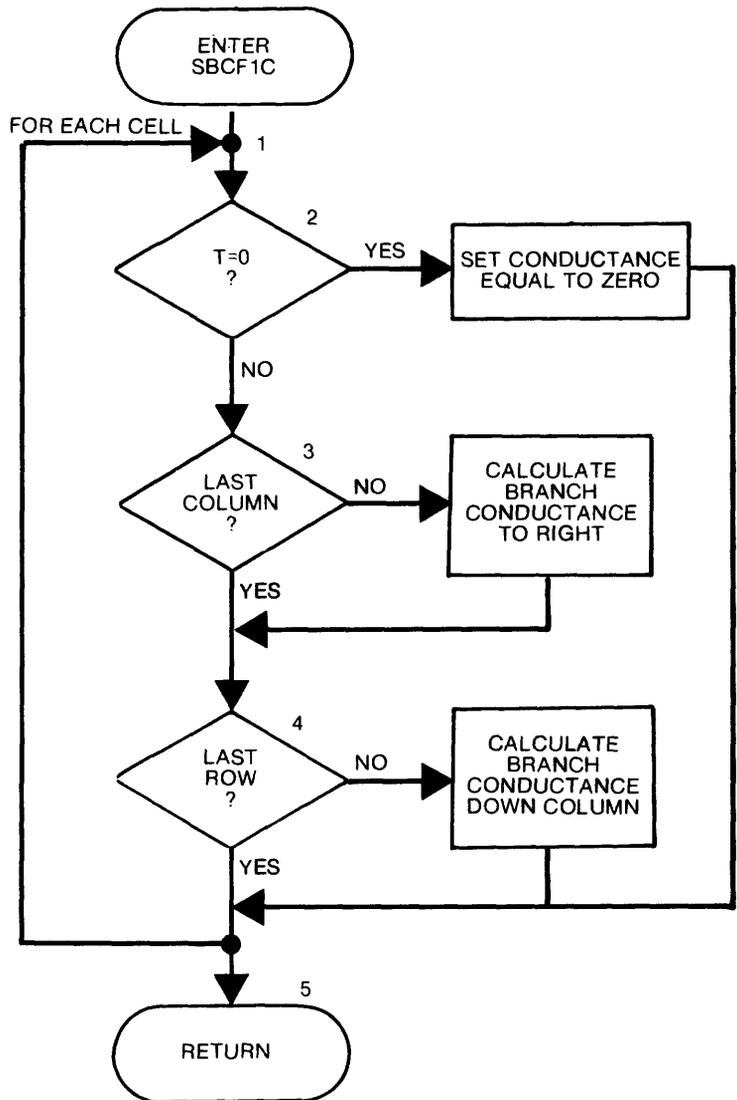
This equation is used to calculate conductances along rows and columns. When calculating conductance along rows,  $L_1$  and  $L_2$  are DELR(J) and DELR(J+1), respectively, and  $W$  is DELC(I). When calculating conductance along columns,  $L_1$  and  $L_2$  are DELC(I) and DELC(I+1), respectively, and  $W$  is DELR(J). Conductance along columns is also multiplied by TRPY(K), the ratio of conductivity in the column direction to conductivity in the row direction in layer K.

1. Process cells one at a time calculating branch conductances from that cell to the one on the right and the one in front.
2. If the transmissivity is equal to zero, set the branch conductance equal to zero and skip to the next cell.
3. If the transmissivity of the cell is not zero and if there is a cell to the right, calculate the branch conductance (CR) along the row.
4. If the transmissivity of the cell is not zero and there is a cell in front, calculate the conductance along the column.
5. RETURN.

Note: Transmissivity, which was temporarily stored in CC, will be lost when conductances are calculated.

CR(I,J,K) contains the conductance  $CR_{i,j+1/2,k}$  between node I,J,K and node I,J+1,K. Node (I,NCOL,K) is on the right side of the grid. Thus there will be no nodes to the right and CR(I,NCOL,K) will be equal to zero. Similarly CC(NROW,J,K) will be equal to zero.

Flow Chart for Module SBCF1C



```

SUBROUTINE SBCF1C(CR,CC,TRPY,DELX,DELC,K,NCOL,NROW,NLAY)
C
C
C-----VERSION 1010 16NOV1982 SBCF1C
C *****
C COMPUTE BRANCH CONDUCTANCE USING HARMONIC MEAN OF BLOCK
C CONDUCTANCES -- BLOCK TRANSMISSIVITY IS IN CC UPON ENTRY
C *****
C
C SPECIFICATIONS:
C -----
C
C DIMENSION CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY)
C 2 , TRPY(NLAY), DELX(NCOL), DELC(NROW)
C -----
C YX=TRPY(K)*2.
C
C1-----FOR EACH CELL CALCULATE BRANCH CONDUCTANCES FROM THAT CELL
C1-----TO THE ONE ON THE RIGHT AND THE IN FRONT.
C DO 40 I=1,NROW
C DO 40 J=1,NCOL
C T1=CC(J,I,K)
C
C2-----IF T=0 THEN SET CONDUCTANCE EQUAL TO 0. GO ON TO NEXT CELL.
C IF(T1.NE.0.) GO TO 10
C CR(J,I,K)=0.
C GO TO 40
C
C3-----IF THIS IS NOT THE LAST COLUMN(RIGHTMOST) THEN CALCULATE
C3-----BRANCH CONDUCTANCE IN THE ROW DIRECTION (CR) TO THE RIGHT.
C 10 IF(J.EQ.NCOL) GO TO 30
C T2=CC(J+1,I,K)
C CR(J,I,K)=2.*T2*T1*DELC(I)/(T1*DELC(J+1)+T2*DELC(J))
C
C4-----IF THIS IS NOT THE LAST ROW(FRONTMOST) THEN CALCULATE
C4-----BRANCH CONDUCTANCE IN THE COLUMN DIRECTION (CC) TO THE FRONT.
C 30 IF(I.EQ.NROW) GO TO 40
C T2=CC(J,I+1,K)
C CC(J,I,K)=YX*T2*T1*DELC(J)/(T1*DELC(I+1)+T2*DELC(I))
C 40 CONTINUE
C
C5-----RETURN
C RETURN
C END

```

List of Variables for Module SBCF1C

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
I	Module	Index for rows.
J	Module	Index for columns.
K	Module	Index for layers.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
TRPY	Package	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.
T1	Module	Temporary field for CC(J,I,K).
T2	Module	Temporary field for CC(J+1,I,K).
YX	Module	TRPY(K)*2.

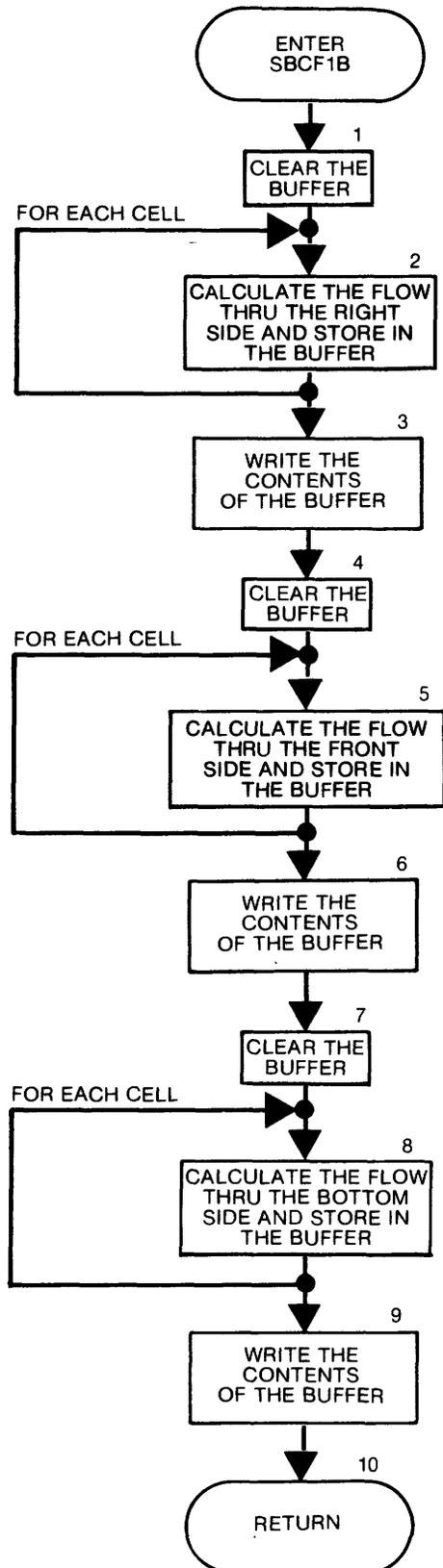
### Narrative for Module SBCF1B

This module calculates flow across cell faces. It is called by module BCF1BD when the user has requested cell-by-cell flow terms. It performs its tasks in the following order:

1. Clear the buffer (BUFF) in which cell-by-cell flow terms are gathered as they are calculated.
2. For each cell, calculate the flow in the row direction through the right face of the cell and store it in the buffer.
3. Call utility module UBUDSV to write the contents of the buffer.
4. Clear the buffer (BUFF) in which cell-by-cell flow terms are gathered as they are calculated.
5. For each cell, calculate the flow in the column direction through the front face of the cell and store it in the buffer.
6. Call utility module UBUDSV to write the contents of the buffer.
7. Clear the buffer (BUFF) in which cell-by-cell flow terms are gathered as they are calculated.
8. For each cell, calculate the flow in the vertical direction through the lower face of the cell and store it in the buffer.
9. Call utility module UBUDSV to write the contents of the buffer.
10. RETURN.

## Flow Chart for Module SBCF1B

**BUFFER:** the buffer is an array with one element for each cell in the grid. It is used to store the results of cell-by-cell calculations until all cells have been processed. The contents of the buffer are then recorded as a unit.



```

SUBROUTINE SBCF1B(HNEW,IBOUND,CR,CC,CV,TOP,NCOL,NROW,NLAY,
1      KSTP,KPER,IBCFCB,BUFF,IOUT)
C
C-----VERSION 1004 03MAY1983 SBCF1B
C
C      *****
C      COMPUTE FLOW ACROSS EACH CELL WALL
C      *****
C
C      SPECIFICATIONS:
C      -----
C      DOUBLE PRECISION HNEW,HD
C
C      DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
1      CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
2      CV(NCOL,NROW,NLAY), TOP(NCOL,NROW,NLAY),
3      BUFF(NCOL,NROW,NLAY)
C
C      COMMON /FLWCOM/LAYCON(80)
C
C      DIMENSION TEXT(12)
C
C      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4),TEXT(5),TEXT(6),TEXT(7),
1      TEXT(8),TEXT(9),TEXT(10),TEXT(11),TEXT(12)
2      /'FLOW',' RIG','HT F','ACE ',
2      'FLOW',' FRO','NT F','ACE ','FLOW',' LOW','ER F','ACE '/
C      -----
C
C      NCM1=NCOL-1
C      IF(NCM1.LT.1) GO TO 405
C
C1-----CLEAR THE BUFFER
C      DO 310 K=1,NLAY
C      DO 310 I=1,NROW
C      DO 310 J=1,NCOL
C      BUFF(J,I,K)=0.
C      310 CONTINUE
C
C2-----FOR EACH CELL CALCULATE FLOW THRU RIGHT FACE & STORE IN BUFFER
C      DO 400 K=1,NLAY
C      DO 400 I=1,NROW
C      DO 400 J=1,NCM1
C      IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J+1,I,K).LE.0)) GO TO 400
C      HDIFF=HNEW(J,I,K)-HNEW(J+1,I,K)
C      BUFF(J,I,K)=HDIFF*CR(J,I,K)
C      400 CONTINUE
C
C3-----RECORD CONTENTS OF BUFFER
C      CALL UBUDSV(KSTP,KPER,TEXT(1),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C4-----CLEAR THE BUFFER
C      405 NRM1=NROW-1
C      IF(NRM1.LT.1) GO TO 505
C      DO 410 K=1,NLAY

```

```

        DO 410 I=1,NROW
        DO 410 J=1,NCOL
        BUFF(J,I,K)=0.
410 CONTINUE
C
C5-----FOR EACH CELL CALCULATE FLOW THRU FRONT FACE & STORE IN BUFFER
        DO 500 K=1,NLAY
        DO 500 I=1,NRM1
        DO 500 J=1,NCOL
        IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J,I+1,K).LE.0)) GO TO 500
        HDIFF=HNEW(J,I,K)-HNEW(J,I+1,K)
        BUFF(J,I,K)=HDIFF*CC(J,I,K)
500 CONTINUE
C
C6-----RECORD CONTENTS OF BUFFER.
        CALL UBUDSV(KSTP,KPER,TEXT(5),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
505 NLM1=NLAY-1
        IF(NLM1.LT.1) GO TO 1000
C
C7-----CLEAR THE BUFFER
        DO 510 K=1,NLAY
        DO 510 I=1,NROW
        DO 510 J=1,NCOL
        BUFF(J,I,K)=0.
510 CONTINUE
C
C8-----FOR EACH CELL CALCULATE FLOW THRU LOWER FACE & STORE IN BUFFER
        KT=0
        DO 600 K=1,NLM1
        IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
        DO 600 I=1,NROW
        DO 600 J=1,NCOL
        IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J,I,K+1).LE.0)) GO TO 600
        HD=HNEW(J,I,K+1)
        IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 580
        TMP=HD
        IF(TMP.LT.TOP(J,I,KT+1)) HD=TOP(J,I,KT+1)
580 HDIFF=HNEW(J,I,K)-HD
        BUFF(J,I,K)=HDIFF*CV(J,I,K)
600 CONTINUE
C
C9-----RECORD CONTENTS OF BUFFER.
        CALL UBUDSV(KSTP,KPER,TEXT(9),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C10----RETURN
1000 RETURN
        END

```

List of Variables for Module SBCF1B

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
HD	Module	Temporary field for head.
HDIFF	Module	Head difference between two adjacent nodes.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
I	Module	Index for rows.
ICBCFB	Package	Flag and a unit number. > 0, unit number on which the cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will be not be printed or recorded < 0, flow from each constant-head cell will be printed whenever ICBCFL is set.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
J	Module	Index for columns.
K	Module	Index for layers.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
KT	Module	Index for tops of layers.
LAYCON	Package	DIMENSION(80), Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant) 3 - Layer confined/unconfined (transmissivity varies).
NCM1	Module	NCOL-1.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NLM1	Module	NLAY-1.
NRM1	Module	NROW-1.
NROW	Global	Number of rows in the grid.
TEXT	Module	Label to be printed or recorded with array data.
TMP	Module	Temporary field for head.
TOP	Package	DIMENSION (NCOL,NROW,NTOP), Elevation of top of layers. (NTOP is number of layers for which LAYCON = 2 or 3.)

### Narrative for Module SBCF1F

This module calculates flow from constant-head cells. The flows are accumulated by sign to get flow into (CHIN) and out of (CHOUT), the flow field for inclusion in the overall volumetric budget. The flows are also accumulated by cell to get the total flow from each constant-head cell on a cell-by-cell basis. Module SBCF1F is called by module BCF1BD and calls utility module UBUDSV.

Module SBCF1F performs its functions in the following order:

1. Clear the fields CHIN and CHOUT in which flow into and out of the flow field, respectively, will be accumulated.
2. If cell-by-cell flow terms will be recorded, clear the buffer (BUFF) in which they will be stored as they are calculated.
3. For each cell, calculate the flow to and from constant-head cells.  
DO STEPS 4-12.
4. If the cell is not a constant-head cell, skip further processing and go on to the next cell.
5. Clear the six fields corresponding to the six faces through which the flows will be calculated.
6. For each face of the cell, calculate the flow out of the cell through that face (STEPS 7-11).
7. If there is not a variable-head cell which shares the face, go on to the next face.

8. Calculate the flow through the face into the adjacent cell.
9. Test the sign of the flow to see if it is positive (into the adjacent variable-head cell from the constant-head cell) or negative (out of the adjacent variable-head cell into the constant-head cell). GO TO EITHER STEP 10 OR 11.
10. If the sign is negative, add the flow rate to CHOUT (flow out of the flow domain).
11. If the sign is positive, add the flow rate to CHIN (flow out of the flow domain).
12. Add together the flow terms ( $x_1, x_2, x_3, x_4, x_5, x_6$ ) corresponding to the six faces and leave in the field RATE.
13. If the user specified a negative number for IBCFCB, and ICBCFL  $\neq 0$ , print the flows (RATE) from the constant-head cell into the aquifer.
14. If the cell-by-cell terms are to be recorded, add the six flow rates out of the cell and store them in the buffer until all cells are finished.
15. If the cell-by-cell terms are to be recorded, call utility module UBUDSV to record them.
16. Put flow rates, into and out of the flow domain from constant-head cells, into the VBVL array for inclusion in the overall volumetric budget. Put labels for those budget terms into VBNM.
17. RETURN.

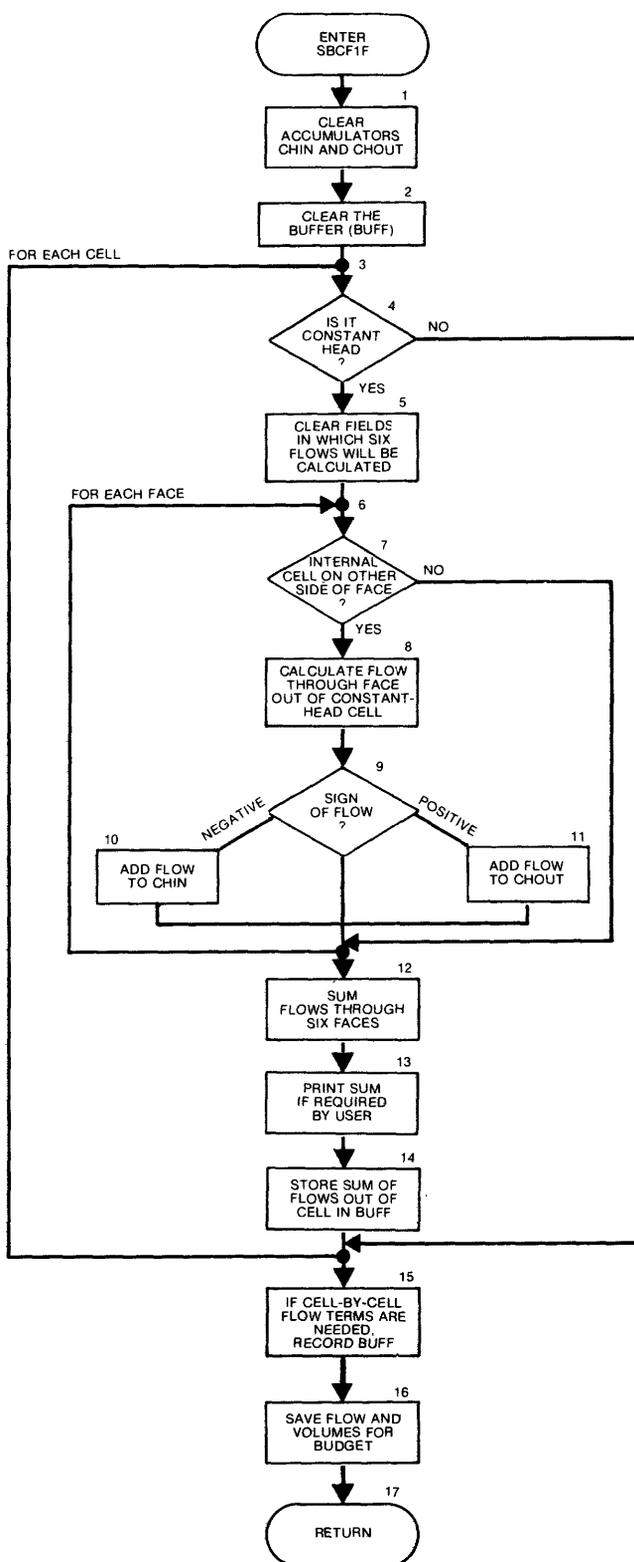
## Flow Chart for Module SBCF1F

CHIN is a field in which flows, into the flow domain from constant-head cells, will be accumulated.

CHOUT is a field in which flows, out of the flow domain to constant-head cells, will be accumulated.

BUFF is a buffer in which cell-by-cell flow terms will be stored as they are calculated prior to recording them on disk.

INTERNAL CELLS are those in which head varies. They are in opposition to EXTERNAL CELLS (inactive or constant head) which are on or outside of a boundary.



```

SUBROUTINE SBCF1F(VBNM,VBVL,MSUM,HNEW,IBOUND,CR,CC,CV,
1  TOP,DELT,NCOL,NROW,NLAY,KSTP,KPER,IBD,IBCFCB,ICBCFL,
2  BUFF,IOUT)
C-----VERSION 1123 29MAY1983 SBCF1F
C
C *****
C COMPUTE FLOW FROM CONSTANT HEAD NODES
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW,HD
C
C DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
1  CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
2  CV(NCOL,NROW,NLAY), VBNM(4,20), VBVL(4,20),
3  TOP(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
C
C COMMON /FLWCOM/LAYCON(80)
C
C DIMENSION TEXT(4)
C
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' C','ONST','ANT ','HEAD'/
C -----
C1-----CLEAR BUDGET ACCUMULATORS
C      CHIN=0.
C      CHOUT=0.
C
C2-----CLEAR BUFFER IF CELL-BY-CELL FLOW TERM FLAG(IBD) IS SET
C      IF (IBD.EQ.0) GO TO 8
C      DO 5 K=1,NLAY
C      DO 5 I=1,NROW
C      DO 5 J=1,NCOL
C      BUFF(J,I,K)=0.
C      5 CONTINUE
C
C3-----FOR EACH CELL IF IT IS CONSTANT HEAD COMPUTE FLOW ACROSS 6
C3-----FACES.
C      8 KT=0
C      DO 200 K=1,NLAY
C      LC=LAYCON(K)
C      IF(LC.EQ.3 .OR. LC.EQ.2) KT=KT+1
C      DO 200 I=1,NROW
C      DO 200 J=1,NCOL
C
C4-----IF CELL IS NOT CONSTANT HEAD SKIP IT & GO ON TO NEXT CELL.
C      IF (IBOUND(J,I,K).GE.0)GO TO 200
C
C5-----CLEAR FIELDS FOR SIX FLOW RATES.
C      X1=0.
C      X2=0.
C      X3=0.
C      X4=0.
C      X5=0.
C      X6=0.
C6-----FOR EACH FACE OF THE CELL CALCULATE FLOW THROUGH THAT FACE
C6-----OUT OF THE CONSTANT HEAD CELL AND INTO THE FLOW DOMAIN.
C6-----COMMENTS 7-11 APPEAR ONLY IN THE SECTION HEADED BY COMMENT 6A
C6-----BUT THEY APPLY IN A SIMILAR MANNER TO THE SECTIONS HEADED

```

```

C6-----BY COMMENTS 6B-6F.
C
C6A----CALCULATE FLOW THROUGH THE LEFT FACE
C
C7-----IF THERE IS NOT A VARIABLE HEAD CELL ON THE OTHER SIDE OF THIS
C7-----FACE THEN GO ON TO THE NEXT FACE.
        IF(J.EQ.1) GO TO 30
        IF(IBOUND(J-1,I,K).LE.0)GO TO 30
        HDIFF=HNEW(J,I,K)-HNEW(J-1,I,K)
C
C8-----CALCULATE FLOW THROUGH THIS FACE INTO THE ADJACENT CELL.
        X1=HDIFF*CR(J-1,I,K)
C
C9-----TEST TO SEE IF FLOW IS POSITIVE OR NEGATIVE
        IF (X1) 10,30,20
C
C10----IF NEGATIVE ADD TO CHOUT(FLOW OUT OF DOMAIN TO CONSTANT HEAD).
        10 CHOUT=CHOUT-X1
        GO TO 30
C
C11----IF POSITIVE ADD TO CHIN(FLOW INTO DOMAIN FROM CONSTANT HEAD).
        20 CHIN=CHIN+X1
C
C6B----CALCULATE FLOW THROUGH THE RIGHT FACE
        30 IF(J.EQ.NCOL) GO TO 60
        IF(IBOUND(J+1,I,K).LE.0) GO TO 60
        HDIFF=HNEW(J,I,K)-HNEW(J+1,I,K)
        X2=HDIFF*CR(J,I,K)
        IF(X2)40,60,50
        40 CHOUT=CHOUT-X2
        GO TO 60
        50 CHIN=CHIN+X2
C
C6C----CALCULATE FLOW THROUGH THE BACK FACE.
        60 IF(I.EQ.1) GO TO 90
        IF (IBOUND(J,I-1,K).LE.0) GO TO 90
        HDIFF=HNEW(J,I,K)-HNEW(J,I-1,K)
        X3=HDIFF*CC(J,I-1,K)
        IF(X3) 70,90,80
        70 CHOUT=CHOUT-X3
        GO TO 90
        80 CHIN=CHIN+X3
C
C6D----CALCULATE FLOW THROUGH THE FRONT FACE.
        90 IF(I.EQ.NROW) GO TO 120
        IF(IBOUND(J,I+1,K).LE.0) GO TO 120
        HDIFF=HNEW(J,I,K)-HNEW(J,I+1,K)
        X4=HDIFF*CC(J,I,K)
        IF (X4) 100,120,110
        100 CHOUT=CHOUT-X4
        GO TO 120
        110 CHIN=CHIN+X4
C
C6E----CALCULATE FLOW THROUGH THE UPPER FACE
        120 IF(K.EQ.1) GO TO 150
        IF (IBOUND(J,I,K-1).LE.0) GO TO 150
        HD=HNEW(J,I,K)
        IF(LC.NE.3 .AND. LC.NE.2) GO TO 122
        TMP=HD
        IF(TMP.LT.TOP(J,I,KT)) HD=TOP(J,I,KT)

```

```

122 HDIFF=HD-HNEW(J,I,K-1)
    X5=HDIFF*CV(J,I,K-1)
    IF(X5) 130,150,140
130 CHOUT=CHOUT-X5
    GO TO 150
140 CHIN=CHIN+X5
C
C6F----CALCULATE FLOW THROUGH THE LOWER FACE.
150 IF(K.EQ.NLAY) GO TO 180
    IF(IBOUND(J,I,K+1).LE.0) GO TO 180
    HD=HNEW(J,I,K+1)
    IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 152
    TMP=HD
    IF(TMP.LT.TOP(J,I,KT+1)) HD=TOP(J,I,KT+1)
152 HDIFF=HNEW(J,I,K)-HD
    X6=HDIFF*CV(J,I,K)
    IF(X6) 160,180,170
160 CHOUT=CHOUT-X6
    GO TO 180
170 CHIN=CHIN+X6
C
C12-----SUM UP FLOWS THROUGH SIX SIDES OF CONSTANT HEAD CELL.
180 RATE=X1+X2+X3+X4+X5+X6
C
C13-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IBCFCB<0).
    IF(IBCFCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
1   KPER,KSTP,K,I,J,RATE
900 FORMAT(1H0,4A4,' PERIOD',I3,' STEP',I3,' LAYER',I3,
1   ' ROW',I4,' COL',I4,' RATE ',G15.7)
C
C14----IF CELL-BY-CELL FLAG SET STORE SUM OF FLOWS FOR CELL IN BUFFER
    IF(IBD.EQ.1) BUFF(J,I,K)=RATE
C
200 CONTINUE
C
C15----IF CELL-BY-CELL FLAG SET THEN RECORD CONTENTS OF BUFFER
    IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT(1),
1   IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C16----SAVE TOTAL CONSTANT HEAD FLOWS AND VOLUMES IN VBVL TABLE
C16----FOR INCLUSION IN BUDGET. PUT LABELS IN VBNM TABLE.
    VBVL(1,MSUM)=VBVL(1,MSUM)+CHIN*DELT
    VBVL(2,MSUM)=VBVL(2,MSUM)+CHOUT*DELT
    VBVL(3,MSUM)=CHIN
    VBVL(4,MSUM)=CHOUT
C
C    ---SETUP VOLUMETRIC BUDGET NAMES
    VBNM(1,MSUM)=TEXT(1)
    VBNM(2,MSUM)=TEXT(2)
    VBNM(3,MSUM)=TEXT(3)
    VBNM(4,MSUM)=TEXT(4)
C
    MSUM=MSUM+1
C
C
C17----RETURN
    RETURN
    END

```

List of Variables for Module SBCF1F

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CHIN	Module	Accumulator for flow into the model area from constant heads.
CHOUT	Module	Accumulator for flow out of the model area to constant heads.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELT	Global	Length of the current time step.
HD	Module	Temporary field containing a value from HNEW.
HDIFF	Module	Head difference between one node and the adjacent node.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
I	Module	Index for rows.
ICBCFB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, flow from each constant-head cell will be printed whenever ICBCFL is set.
IBD	Package	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
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 (flow to constant heads) will be either printed or recorded for the current time step.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
J	Module	Index for columns.
K	Module	Index for layers.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.

List of Variables for Module SBCF1F (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KT LAYCON	Module Package	Index for tops of layers. DIMENSION(80), Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
LC	Module	Temporary label for an element of LAYCON.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RATE	Module	Flow from the constant-head cell into the aquifer. (Reverse the sign to get the flow from the aquifer into the constant-head cell.)
TEXT	Module	Label to be printed or recorded with array data.
TMP	Module	Temporary field for head.
TOP	Package	DIMENSION (NCOL,NROW,NTOP), Elevation of top of layers. (NTOP is the number of layers for which LAYCON = 2 or 3.)
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the 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.
X1	Module	Flow through the left face.
X2	Module	Flow through the right face.
X3	Module	Flow through the back face.
X4	Module	Flow through the front face.
X5	Module	Flow through the upper face.
X6	Module	Flow through the lower face.

CHAPTER 6  
RIVER PACKAGE

Conceptualization and Implementation

Rivers may contribute water to the aquifer or drain water from the aquifer depending on the head gradient between the river and the aquifer. The effect of leakage through the riverbed on the shape of the water table is shown in figure 32. The purpose of the River Package is to simulate the effect of that leakage.

To simulate the effect of river leakage in the model, terms representing the leakage are added to the ground-water flow equation (eq. 27) for each cell. The river is divided into reaches, each of which is completely contained in a single cell (fig. 33). River/aquifer leakage is defined between each river reach and the model cell that contains the reach.

Figure 34(a) shows a cross section of a grid cell containing a river reach. The riverbed has been exaggerated to illustrate that water must pass through the bed to get from the river into the aquifer cell. Figure 34(b) is a block diagram of the same situation; the riverbed is represented by a rectilinear prism of homogeneous porous material.

Leakage through a reach of riverbed (fig. 35) is approximated by Darcy's law as

$$QRIV = KLW(HRIV - HAQ)/M \quad (63)$$

where

QRIV is the leakage through the reach of the riverbed ( $L^3t^{-1}$ );

K is the hydraulic conductivity of the riverbed ( $Lt^{-1}$ );

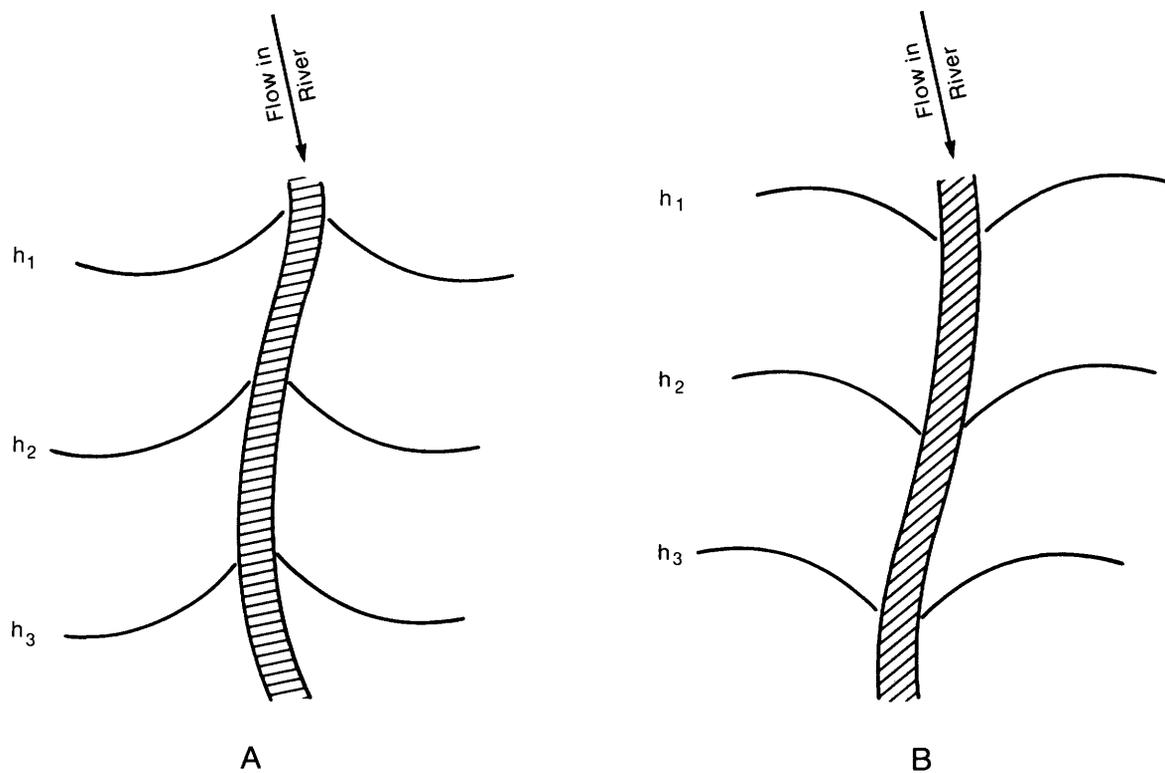


Figure 32.—Water table contours near (a) a gaining stream and (b) a losing stream.

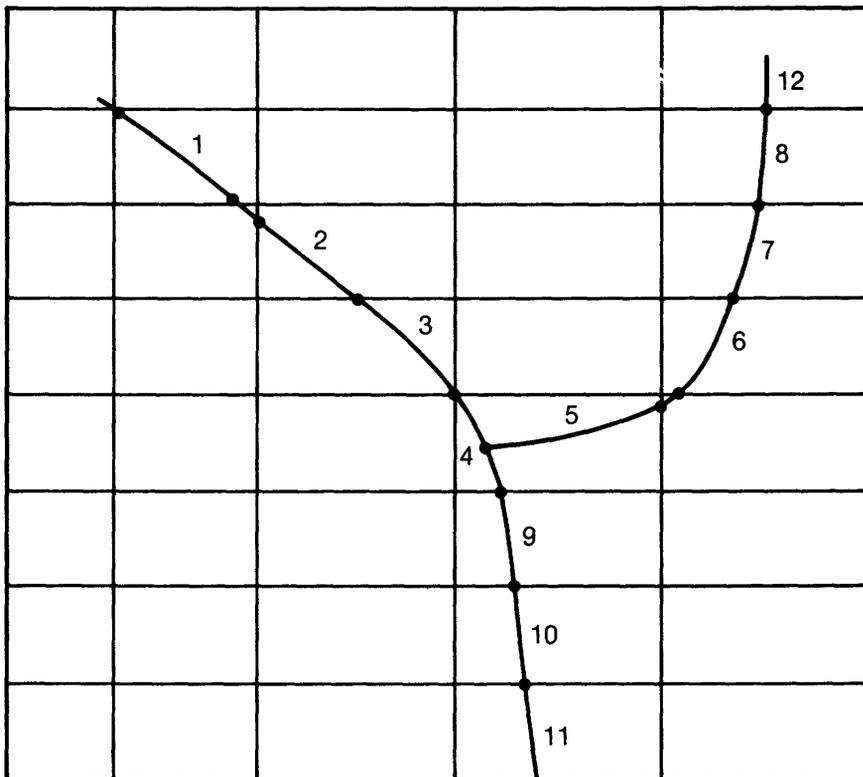


Figure 33.—Discretization of a river into reaches. Some small reaches are ignored.

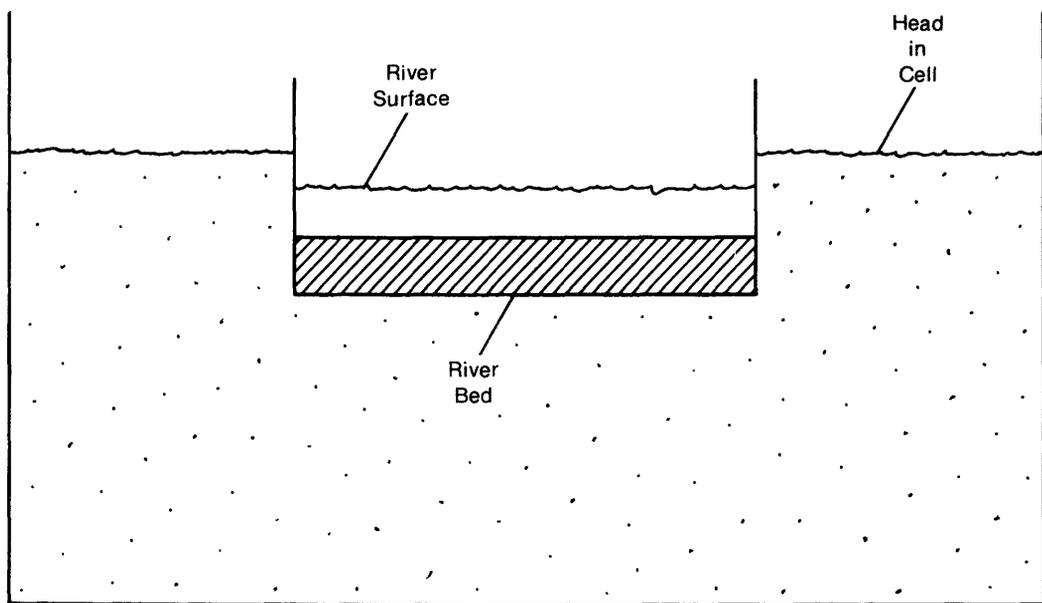
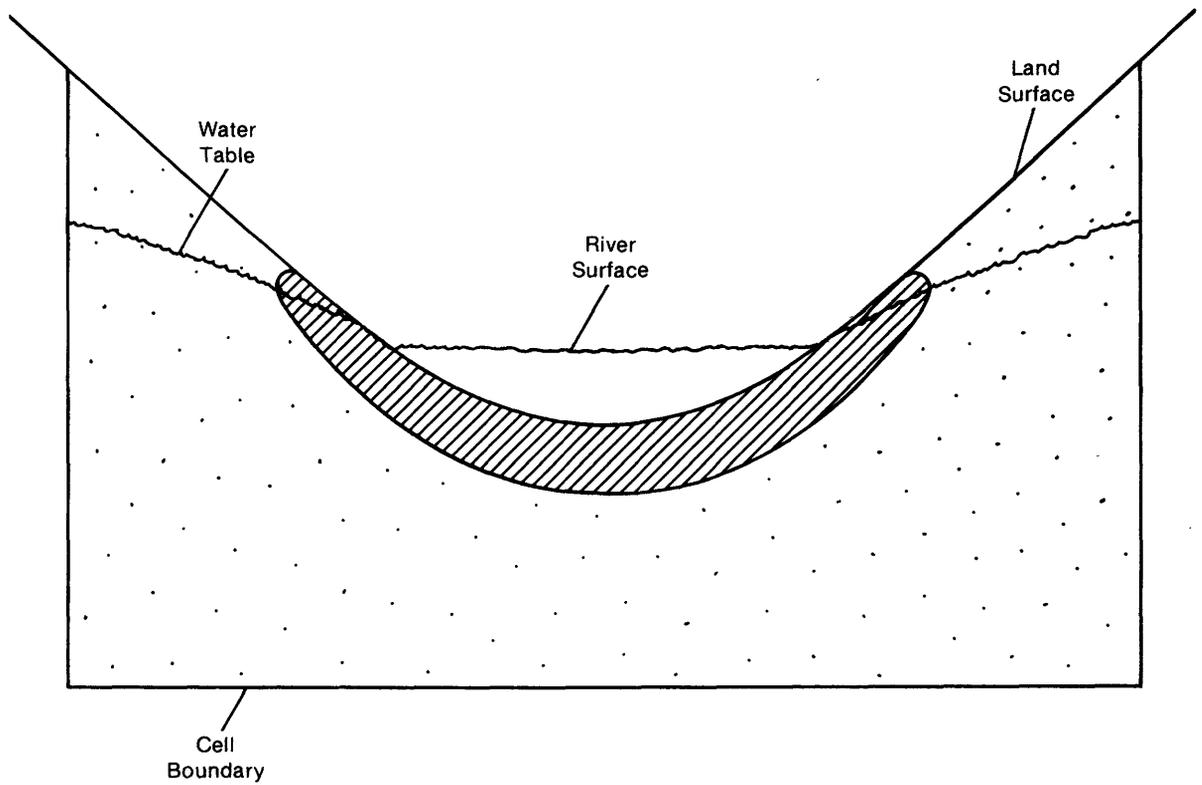


Figure 34.—(a) Cross section of an aquifer containing a river and  
 (b) Block diagram of a cross section of an aquifer containing  
 a river.

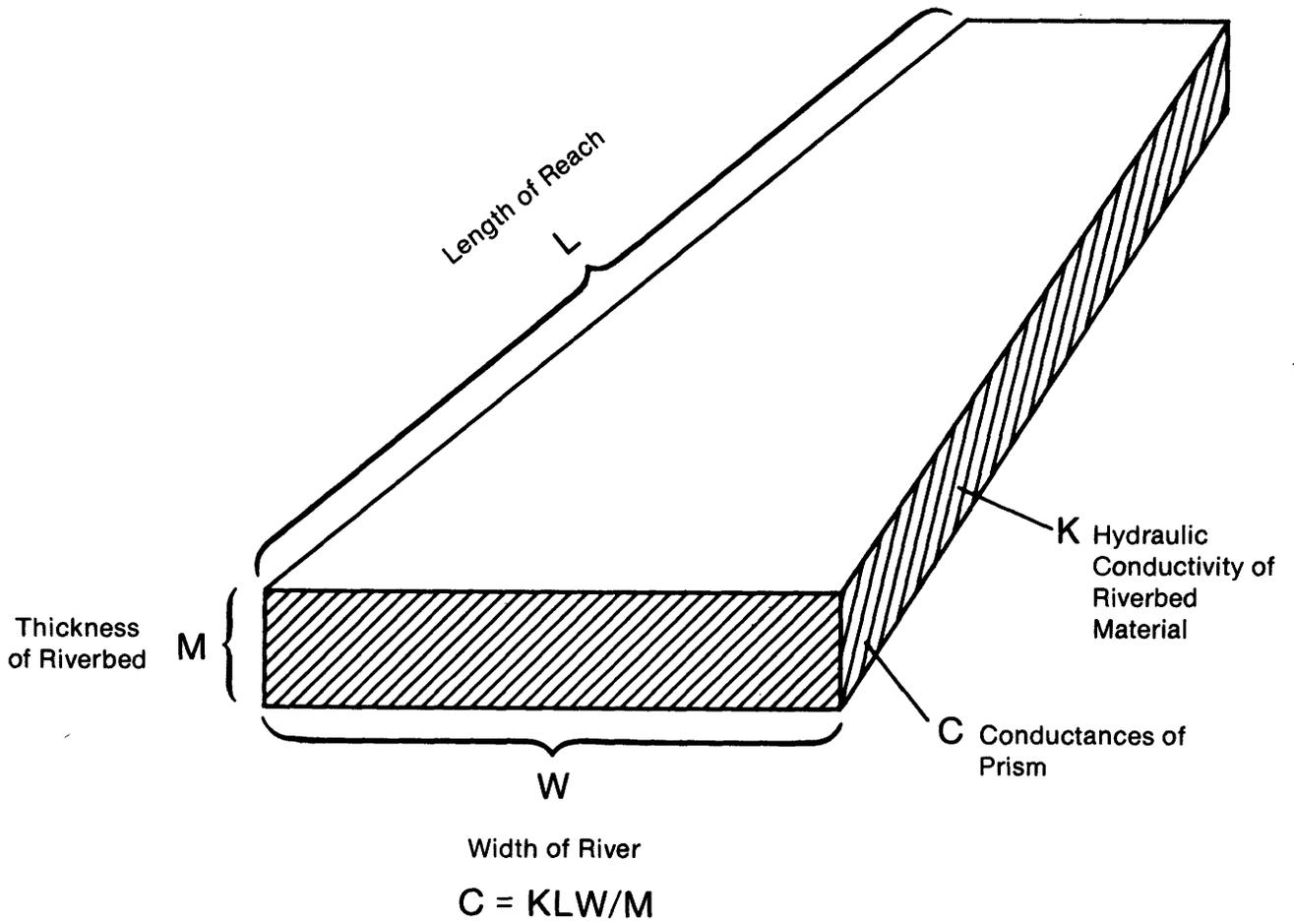


Figure 35.—A riverbed viewed as a prism of porous material.

L is the length of the reach (L);

W is the width of the river (L);

M is the thickness of the riverbed (L);

HAQ is the head on the aquifer side of the riverbed (L); and

HRIV is the head on the river side of the riverbed (L).

Equation 63 can be rewritten in terms of conductance of the reach of the riverbed as

$$QRIV = CRIV(HRIV - HAQ) \quad (64)$$

where

CRIV is the conductance of the reach of the riverbed ( $CRIV = KLW/M$ ).

The head on the river side of the riverbed is the river stage (head in the river). The head on the aquifer side of the riverbed is slightly more complex. Figure 36 shows a situation in which the porous material adjacent to the riverbed is fully saturated; the head on the aquifer side of the riverbed (HAQ) is equal to the head in the cell. Thus equation 64 can be written

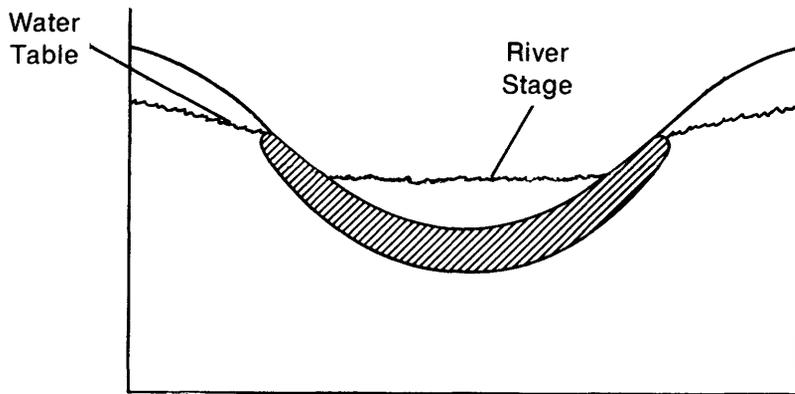
$$QRIV = CRIV(HRIV - H) \quad (65)$$

where

H is the head in the cell (L).

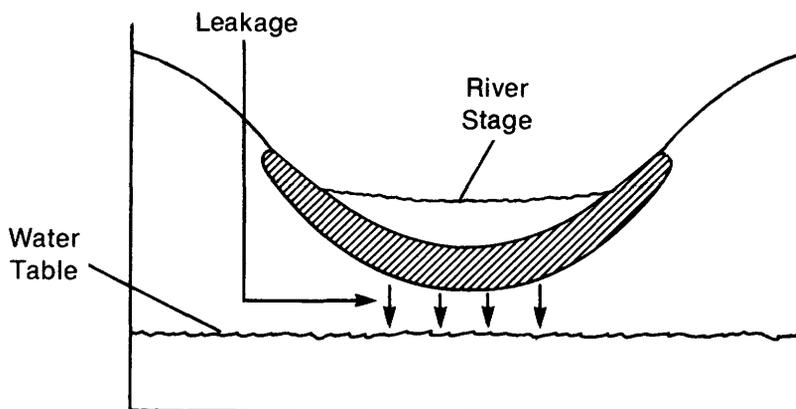
If, however, the material adjacent to the riverbed is not saturated (fig. 36), the head on the aquifer side of the riverbed is equal to the elevation of the bottom of the riverbed (R BOT). In that case, equation 64 can be written

$$QRIV = CRIV(HRIV - R BOT). \quad (66)$$



Head on the Aquifer Side of the River Is Equal to Head in the Cell

A



Head on the Aquifer Side of the River Is Equal to Elevation of Bottom of Riverbed

B

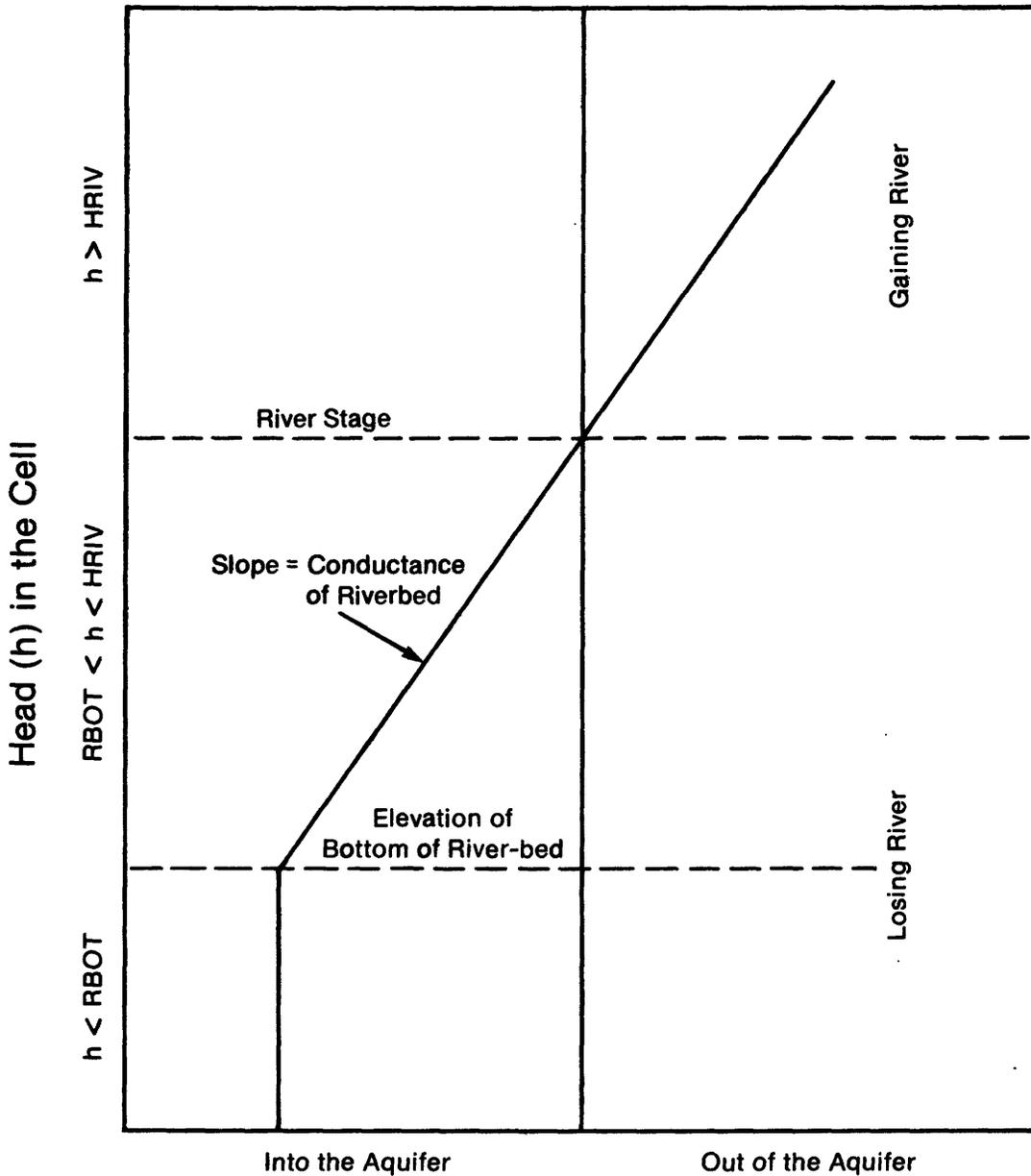
Figure 36.—Cross section showing the relation between head on the aquifer side of the riverbed and head in the cell. Head in the cell is equal to the water-table elevation.

The choice of equation 65 or 66 to determine leakage depends on the head in the cell. The relationship between river leakage and head in the cell is shown in figure 37. The simple model of river leakage represented by the graph in figure 37 is based on the assumption that leakage from the river is independent of the location of the river within the cell. Thus, although there may be two river reaches in one cell, they are both assumed to be at the node. The model has also assumed that there is always enough water in the river to supply the aquifer; the user should compare leakage rates with river discharge rates and insure that they are in agreement.

Data describing each river reach is stored in a list (RIVR) and is specified by the user for each stress period. Input consists of one record for each river reach, which specifies the cell containing the reach (layer, row, and column) and the three parameters needed to calculate seepage--river stage, riverbed conductance, and riverbed bottom elevation.

At the start of each iteration, terms representing river seepage are added to the flow equation. For each river reach, the appropriate river seepage equation is added to the flow equation for the cell containing the reach. The choice of which river seepage equation to use, equation 65 or equation 66, is made by comparing the most recent value of HNEW at the cell to RBOT for the reach. Since this process is done at the start of each iteration, the most current value of HNEW is the value from the previous iteration. Thus, the check for which river seepage equation to use lags behind the seepage calculations by one iteration.

If equation 65 is selected, the term  $-CRIV$  is added to the term HCOF and the term  $-CRIV*HRIV$  is added to RHS. If equation 66 is selected, the term  $-CRIV (HRIV - RBOT)$  is added to the term RHS; nothing is added to HCOF.



### Leakage Through the Riverbed

#### Explanation

HRIV Head (Stage) in the River

RBOT Elevation of the Bottom of the Riverbed

Figure 37.—Leakage through a riverbed into an aquifer as a function of head in the aquifer.

## River Package Input

Input to the River (RIV) Package is read from the unit specified in IUNIT(4).

FOR EACH SIMULATION

RIV1AL

1. Data: MXRIVR IRIVCB  
Format: I10 I10

FOR EACH STRESS PERIOD

RIV1RP

2. Data: ITMP  
Format: I10

3. Data: Layer Row Column Stage Cond Rbot  
Format: I10 I10 I10 F10.0 F10.0 F10.0

(Input item 3 normally consists of one record for each river reach. If ITMP is negative or zero, item 3 is not read.)

### Explanation of Fields Used in Input Instructions

MXRIVR--is the maximum number of river reaches active at one time.

IRIVCB--is a flag and a unit number.

If IRIVCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If IRIVCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IRIVCB < 0, river leakage for each reach will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

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

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

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

Stage--is the head in the river.

Cond--is the riverbed hydraulic conductance.

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

SAMPLE INPUT TO THE RIVER PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{MXRIVR, IRIVCB}	
2	{ITMP} FOR FIRST STRESS PERIOD	3
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR FIRST REACH	55
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR SECOND REACH	6
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR THIRD REACH	7
2	{ITMP} FOR SECOND STRESS PERIOD	2
2	{ITMP} FOR THIRD STRESS PERIOD	2
2	{ITMP} FOR FOURTH STRESS PERIOD	2
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR FIRST REACH	5
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR SECOND REACH	6
2	{ITMP} FOR FIFTH STRESS PERIOD	2
2	{ITMP} FOR SIXTH STRESS PERIOD	0

## Module Documentation for the River Package

The River Package (RIV1) consists of four modules, all of which are called by the MAIN program. The modules are:

- RIV1AL      Allocates space for a list (RIVR) which will contain an entry for each river reach. Each entry will consist of the location of the cell containing the reach, riverhead, conductance of the riverbed, and the elevation of the bottom of the riverbed.
- RIV1RP      Reads, for each river reach, the location of the cell containing the reach, riverhead, conductance of the riverbed, and elevation of the bottom of the riverbed.
- RIV1FM      Adds for each river reach, the appropriate terms to the accumulators HCOF(I,J,K) and RHS(I,J,K).
- RIV1BD      Calculates the rates and accumulated volume of river leakage into and out of the flow system.

### Narrative for Module RIVIAL

This module allocates space in the X array to store the list of river reaches.

1. Print a message identifying the package and initialize NRIVER (number of river reaches).

2. Read and print MXRIVR (the maximum number of river reaches) and IRIVCB (the unit number for saving cell-by-cell flow terms or a flag indicating whether cell-by-cell flow terms should be printed).

3. Set LCRIVR, which will point to the first element in the river list (RIVR), equal to ISUM, which is currently pointing to the first unallocated element in the X array.

4. Calculate the amount of space needed for the river list (six values for each reach--row, column, layer, riverhead, riverbed conductance, and riverbed bottom elevation) and add it to ISUM.

5. Print the number of elements in the X array used by the River Package.

6. RETURN.

Flow Chart for Module RIV1AL

NRIVER is the number of river reaches being simulated at any given time.

MXRIVR is the maximum number of river reaches simulated.

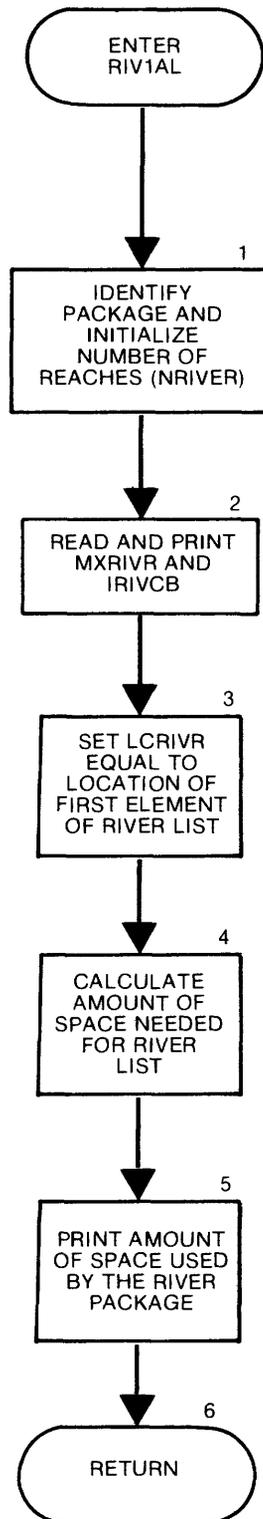
IRIVCB is a flag and a unit number.

If IRIVCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see RIV1BD module) is set.

If IRIVCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IRIVCB < 0, river leakage for each reach will be printed whenever ICBCFL is set.

LCRIVR is the location in the X array of the list of river data (RIVR).



```

SUBROUTINE RIVIAL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,IN,IOUT,
1          IRIVCB)
C
C-----VERSION 0935 08DEC1983 RIVIAL
C *****
C ALLOCATE ARRAY STORAGE FOR RIVERS
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----IDENTIFY PACKAGE AND INITIALIZE NRIVER.
WRITE(IOUT,1)IN
1 FORMAT(1H0,'RIV1 -- RIVER PACKAGE, VERSION 1, 12/08/83',
2' INPUT READ FROM UNIT',I3)
NRIVER=0
C
C2-----READ & PRINT MXRIVR & IRIVCB(UNIT OR FLAG FOR C-B-C FLOWS)
READ(IN,2)MXRIVR,IRIVCB
2 FORMAT(2I10)
WRITE(IOUT,3)MXRIVR
3 FORMAT(1H , 'MAXIMUM OF',I5, ' RIVER NODES')
IF(IRIVCB.GT.0) WRITE(IOUT,9) IRIVCB
9 FORMAT(1X, 'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
IF(IRIVCB.LT.0) WRITE(IOUT,8)
8 FORMAT(1X, 'CELL-BY-CELL FLOWS WILL BE PRINTED')
C
C3-----SET LCRIVR EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
LCRIVR=ISUM
C
C4-----CALCULATE AMOUNT OF SPACE USED BY RIVER LIST.
ISP=6*MXRIVR
ISUM=ISUM+ISP
C
C5-----PRINT AMOUNT OF SPACE USED BY RIVER PACKAGE.
WRITE (IOUT,4)ISP
4 FORMAT(1X,I6, ' ELEMENTS IN X ARRAY ARE USED FOR RIVERS')
ISUM1=ISUM-1
WRITE(IOUT,5)ISUM1,LENX
5 FORMAT(1X,I6, ' ELEMENTS OF X ARRAY USED OUT OF',I7)
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X, ' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C7-----RETURN
RETURN
END

```

List of Variables for Module RIV1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IRIVCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see RIV1BD module) is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, river leakage for each reach will be printed whenever ICBCFL is set.
ISP	Module	Number of words in the X array allocated by this module.
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	ISUM-1.
LCRIVR	Package	Location in the X array of the first element of array RIVR.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXRIVR	Package	Maximum number of river reaches active at any one time.
NRIVER	Package	Number of river reaches active during the current stress period.

### Narrative for Module RIV1RP

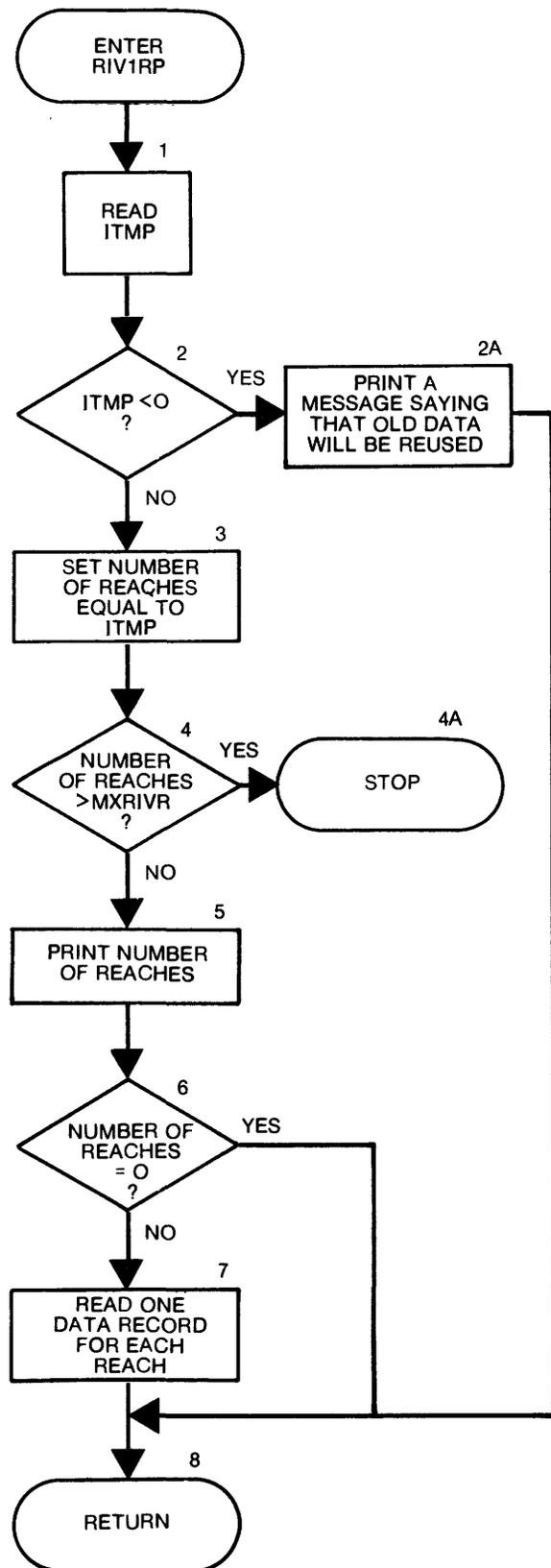
This module reads data to build the river list.

1. Read ITMP. ITMP is the number of river reaches or a flag indicating that river reaches specified for the previous stress period should be reused.
2. Test ITMP. If ITMP is less than zero, the river data read for the last stress period will be reused. Print a message to that effect and RETURN.
3. If ITMP is greater than or equal to zero, it is the number of reaches for this stress period. Set the number of river reaches (NRIVER) in the current stress period equal to ITMP.
4. Compare the number of river reaches (NRIVER) in the current stress period to the number specified as the maximum for the simulation (MXRIVR). If NRIVER is greater than MXRIVR, STOP.
5. Print the number of river reaches in the current stress period (NRIVER).
6. See if there are any river reaches. If there are no river reaches in the current stress period (NRIVER = 0), bypass further river processing.
7. Read and print the layer, row, column, riverhead, riverbed conductance, and the elevation of the bottom of the riverbed for each reach.
8. RETURN.

### Flow Chart for Module RIV1RP

ITMP is both a flag and a counter. If it is greater than or equal to zero, it is the number of reaches to be simulated during the current stress period. If it is less than zero, it indicates that the reaches simulated in the last stress period should be simulated in the current stress period.

MXRIVR is the maximum number of reaches to be simulated.



```

SUBROUTINE RIVIRP(RIVR,NRIVER,MXRIVR,IN,IOUT)
C
C
C-----VERSION 1319 25AUG1982 RIVIRP
C *****
C READ RIVER HEAD, CONDUCTANCE AND BOTTOM ELEVATION
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION RIVR(6,MXRIVR)
C -----
C
C
C1-----READ ITMP(NUMBER OF RIVER REACHES OR FLAG TO REUSE DATA)
      READ(IN,B)ITMP
      8 FORMAT(I10)
C
C2-----TEST ITMP.
      IF(ITMP.GE.0)GO TO 50
C
C2A-----IF ITMP <0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
      7 FORMAT(1H0,'REUSING RIVER REACHES FROM LAST STRESS PERIOD')
      GO TO 260
C
C3-----IF ITMP=> ZERO THEN IT IS THE NUMBER OF RIVER REACHES
      50 NRIVER=ITMP
C
C4-----IF NRIVER>MXRIVR THEN STOP.
      IF(NRIVER.LE.MXRIVR)GO TO 100
      WRITE(IOUT,99)NRIVER,MXRIVR
      99 FORMAT(1H0,'NRIVER(',I4,') IS GREATER THAN MXRIVR(',I4,')')
C
C4A-----ABNORMAL STOP.
      STOP
C
C5-----PRINT NUMBER OF RIVER REACHES IN THIS STRESS PERIOD.
      100 WRITE(IOUT,1)NRIVER
      1 FORMAT(1H0,/,I5,' RIVER REACHES')
C
C6-----IF THERE ARE NO RIVER REACHES THEN RETURN.
      IF(NRIVER.EQ.0) GO TO 260
C
C7-----READ AND PRINT DATA FOR EACH RIVER REACH.
      WRITE(IOUT,3)
      3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X,'COL '
      1,' STAGE CONDUCTANCE BOTTOM ELEVATION RIVER REACH'
      2/1X,15X,80('-'))
      DO 250 II=1,NRIVER
      READ(IN,4)K,I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II)
      4 FORMAT(3I10,3F10.0)
      WRITE(IOUT,5)K,I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II),II
      5 FORMAT(1X,15X,I4,I9,I8,6I3.4,6I4.4,6I9.4,I10)
      RIVR(1,II)=K
      RIVR(2,II)=I
      RIVR(3,II)=J
      250 CONTINUE
C
C8-----RETURN
      260 RETURN
      END

```

List of Variables for Module RIV1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
I	Module	Row number.
II	Module	Index for river reach.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of rivers. ≥ 0, number of rivers active during the current stress period. < 0, same rivers active during the last stress period will be active during the current stress period.
J	Module	Column number.
K	Module	Layer number.
MXRIVR	Package	Maximum number of river reaches active at any one time.
NRIVER	Package	Number of river reaches active during the current stress period.
RIVR	Package	DIMENSION (6,MXRIVR), For each reach: layer, row, column, river head, riverbed conductance and elevation of bottom of riverbed.

### Narrative for Module RIV1FM

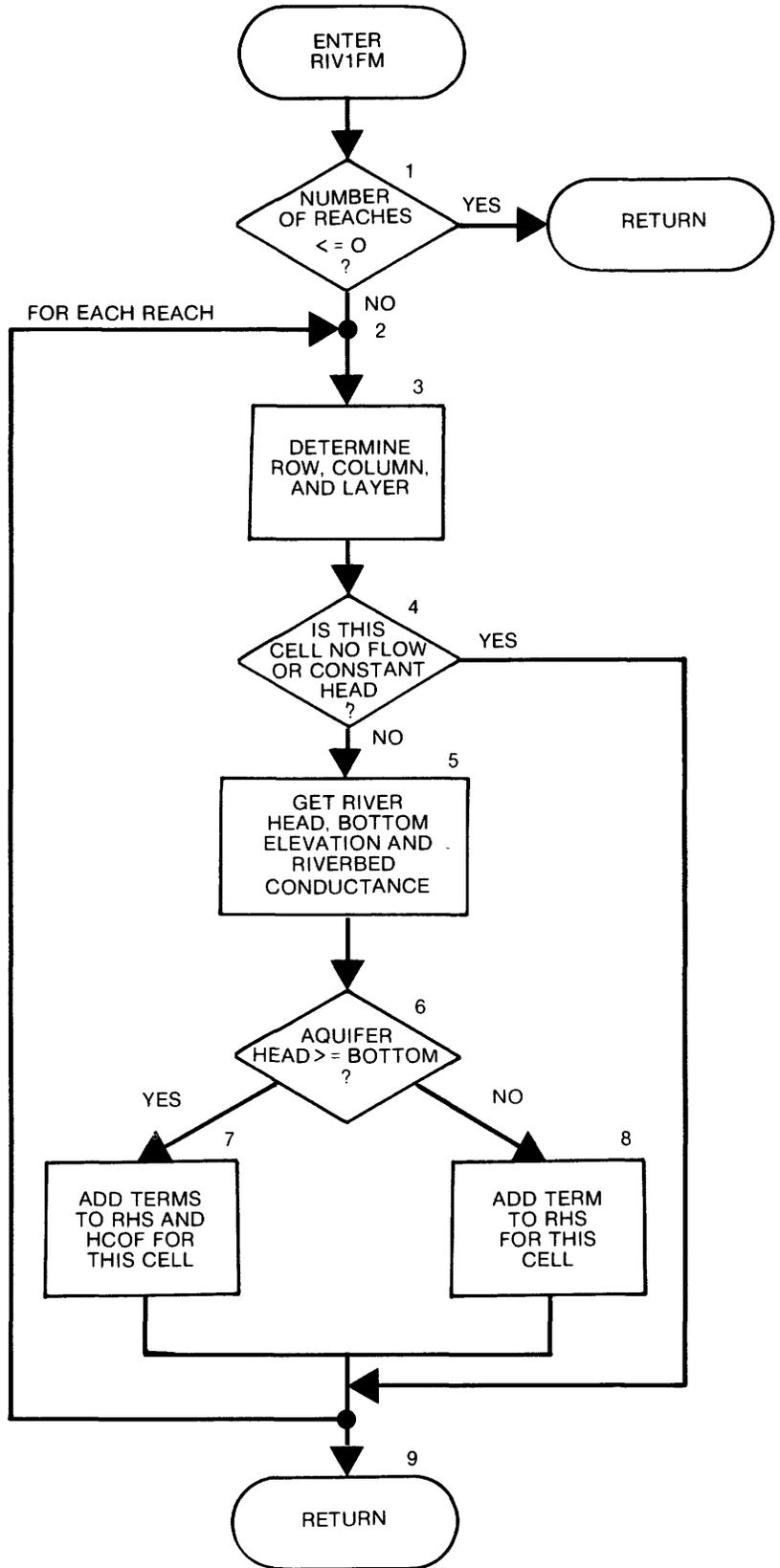
This module adds terms representing river leakage to the accumulators HCOF and RHS.

1. If NRIVER is less than or equal to zero, in the current stress period, there are no river reaches. RETURN.
2. For each reach in the RIVR list, DO STEPS 3-8.
3. Determine the column (IC), row (IR), and layer (IL).
4. If the cell is external ( $IBOUND(IC, IR, IL) \leq 0$ ), bypass processing on this reach and go on to the next reach.
5. Since the cell is internal, get the river data (riverhead conductance of the riverbed and elevation of the bottom of the riverbed).
6. Compare the head in the aquifer (HNEW) to the elevation of the bottom of the riverbed (RBOT).
7. If the head in the aquifer (HNEW) is greater than the elevation of the bottom of the riverbed (RBOT), add the term  $-CRIV*HRIV$  to the accumulator RHS and the term  $-CRIV$  to the accumulator HCOF. (CRIV is the riverbed conductance; HRIV is the riverhead.)
8. If the head in the aquifer (HNEW) is less than or equal to RBOT, add the term  $-CRIV*(HRIV - RBOT)$  to the accumulator RHS.
9. RETURN.

Flow Chart for Module RIV1FM

RHS is an accumulator in which the right hand side of the equation is formulated.

HCOF is an accumulator in which the coefficient of head in the cell is formulated.



```

SUBROUTINE RIV1FM(NRIVER,MXRIVR,RIVR,HNEW,HCOF,RHS,IBOUND,
1          NCOL,NROW,NLAY)
C
C-----VERSION 0915 27AUG1982 RIV1FM
C *****
C ADD RIVER TERMS TO RHS AND HCOF
C *****
C
C SPECIFICATIONS:
C -----
C
C DOUBLE PRECISION HNEW
C DIMENSION RIVR(6,MXRIVR),HNEW(NCOL,NROW,NLAY),
1          HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY),
2          IBOUND(NCOL,NROW,NLAY)
C -----
C
C
C1-----IF NRIVER<=0 THERE ARE NO RIVERS. RETURN.
          IF(NRIVER.LE.0)RETURN
C
C2-----PROCESS EACH CELL IN THE RIVER LIST.
          DO 100 L=1,NRIVER
C
C3-----GET COLUMN, ROW, AND LAYER OF CELL CONTAINING REACH
          IL=RIVR(1,L)
          IR=RIVR(2,L)
          IC=RIVR(3,L)
C
C4-----IF THE CELL IS EXTERNAL SKIP IT.
          IF(IBOUND(IC,IR,IL).LE.0)GO TO 100
C
C5-----SINCE THE CELL IS INTERNAL GET THE RIVER DATA.
          HRIV=RIVR(4,L)
          CRIV=RIVR(5,L)
          RBOT=RIVR(6,L)
          HHNEW=HNEW(IC,IR,IL)
C
C6-----COMPARE AQUIFER HEAD TO BOTTOM OF STREAM BED.
          IF(HHNEW.LE.RBOT)GO TO 96
C
C7-----SINCE HEAD>BOTTOM ADD TERMS TO RHS AND HCOF.
          RHS(IC,IR,IL)=RHS(IC,IR,IL)-CRIV*HRIV
          HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-CRIV
          GO TO 100
C
C8-----SINCE HEAD<BOTTOM ADD TERM ONLY TO RHS.
          96 RHS(IC,IR,IL)=RHS(IC,IR,IL)-CRIV*(HRIV-RBOT)
          100 CONTINUE
C
C9-----RETURN
          RETURN
          END

```

List of Variables for Module RIV1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
CRIV	Module	Riverbed conductance.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HHNEW	Module	HNEW (J,I,K), Single precision.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HRIV	Module	Head in the river.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Column number of the cell containing the river reach.
IL	Module	Layer number of the cell containing the river reach.
IR	Module	Row number of the cell containing the river reach.
L	Module	Index for river reaches.
MXRIVR	Package	Maximum number of river reaches active at any one time.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRIVER	Package	Number of river reaches active during the current stress period.
NROW	Global	Number of rows in the grid.
RBOT	Module	Temporary field: elevation of the river bottom.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.
RIVR	Package	DIMENSION (6,MXRIVR), For each reach: layer, row, column, riverhead, riverbed conductance and elevation of bottom of riverbed.

### Narrative for Module RIV1BD

This module calculates rates and volumes transferred between the aquifer and rivers.

1. Initialize the cell-by-cell flow-term flag (IBD) and the rate accumulators (RATIN and RATOUT).

2. If there are no reaches ( $NRIVER \leq 0$ ), skip down to step 17 and put zeros into the budget terms for rivers.

3. Test to see if the cell-by-cell flow terms are to be saved on the disk. They will not be saved if either of the following conditions hold: (1) This is not the proper time step ( $ICBCFL = 0$ ) or (2) cell-by-cell flow terms are not to be saved for rivers during this simulation ( $IRIVCB \leq 0$ ). If cell-by-cell flow terms will be saved for this package, set the cell-by-cell flow-term flag (IBD) and clear the buffer in which they will be accumulated (BUFF).

4. For each reach, do steps 5-15 accumulating flows from or into the river.

5. Determine the row, column, and layer of the cell containing the reach.

6. If the cell is external ( $IBOUND(I,J,K) \leq 0$ ), bypass further processing of this reach.

7. Get the river parameters from the river list.

8. Check to see if the head in the cell is greater than the elevation of the bottom of the riverbed.

9. If the head in the cell is greater than the elevation of the bottom of the riverbed, set RATE equal to the conductance of the riverbed times the riverhead minus the head in the cell ( $RATE = CRIV*(HRIV - HNEW)$ ).

10. If the head in the cell is less than or equal to the elevation of the bottom of the riverbed, set RATE equal to the conductance of the riverbed times the riverhead minus the elevation of the bottom of the riverbed ( $RATE = CRIV*(HRIV - RBOT)$ ).

11. If the cell-by-cell flow terms are to be printed, print RATE.

12. If the cell-by-cell flow terms are to be saved, add the RATE to the buffer (BUFF).

13. Check to see whether the flow is into or out of the aquifer.

14. If RATE is negative, add it to RATOUT.

15. If RATE is positive, add it to RATIN.

16. See if the cell-by-cell flow terms are to be saved ( $IBD = 1$ ). If they are, call module UBUDSV to record the buffer (BUFF) onto the disk.

17. Move RATIN and RATOUT into the VBVL array for printing by BAS10T. Add RATIN and RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS10T. Move the river budget term labels to VBNM for printing by BAS10T.

18. Increment the budget-term counter (MSUM).

19. RETURN.

## Flow Chart for Module RIV1BD

IBD is a flag which, if set, causes cell-by-cell flow terms for river leakage to be recorded.

EXTERNAL: a cell is said to be external if it is either no flow or constant head (i.e., an equation is not formulated for the cell).

RATE is the leakage rate into the aquifer from the river in a cell.

BUFFER is an array in which values are stored as they are being gathered for printing or recording.

RATOUT is an accumulator to which all flows out of the aquifer are added.

RATIN is an accumulator to which all flows into the aquifer are added.

IRIVCB is a flag and a unit number.

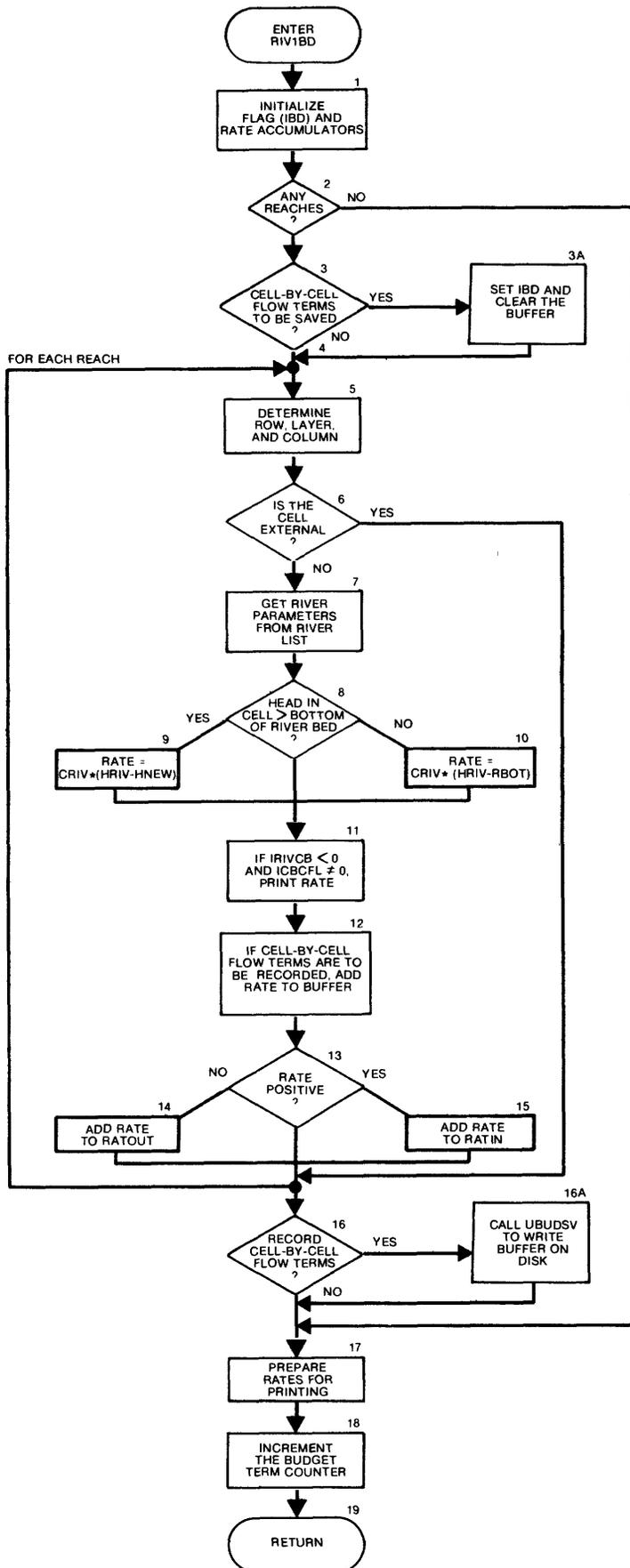
If  $IRIVCB > 0$ , it is the unit number on which cell-by-cell flow terms for rivers will be recorded whenever ICBCFL is set.

If  $IRIVCB = 0$ , cell-by-cell flow terms for rivers will not be printed or recorded.

If  $IRIVCB < 0$ , river leakage for each reach will be printed whenever ICBCFL is set.

ICBCFL is a flag.

If  $ICBCFL \neq 0$ , cell-by-cell flow terms will be printed or recorded (depending on IRIVCB) for the current time step.



```

SUBROUTINE RIV1BD(NRIVER, MXRIVR, RIVR, IBOUND, HNEW,
1      NCOL, NROW, NLAY, DELT, VBVL, VBNM, MSUM, KSTP, KPER, IRIVCB,
2      ICBCFL, BUFF, IOUT)
C-----VERSION 1256 28DEC1983 RIV1BD
C      *****
C      CALCULATE VOLUMETRIC BUDGET FOR RIVERS
C      *****
C
C      SPECIFICATIONS:
C      -----
C      DOUBLE PRECISION HNEW
C      DIMENSION RIVR(6, MXRIVR), IBOUND(NCOL, NROW, NLAY),
1      HNEW(NCOL, NROW, NLAY), VBVL(4, 20), VBNM(4, 20),
2      BUFF(NCOL, NROW, NLAY)
C      DIMENSION TEXT(4)
C      DATA TEXT(1), TEXT(2), TEXT(3), TEXT(4) /' R', 'IVER', ' LEA', 'KAGE' /
C      -----
C
C1-----INITIALIZE CELL-BY-CELL FLOW TERM FLAG (IBD) AND
C1-----ACCUMULATORS (RATIN AND RATOUT).
      IBD=0
      RATIN=0.
      RATOUT=0.
C
C2-----IF NO REACHES KEEP ZEROES IN ACCUMULATORS.
      IF(NRIVER.EQ.0)GO TO 200
C
C3-----TEST TO SEE IF CELL-BY-CELL FLOW TERMS ARE NEEDED.
      IF(ICBCFL.EQ.0 .OR. IRIVCB.LE.0 ) GO TO 10
C
C3A----CELL-BY-CELL FLOW TERMS ARE NEEDED SET IBD AND CLEAR BUFFER.
      IBD=1
      DO 5 IL=1, NLAY
      DO 5 IR=1, NROW
      DO 5 IC=1, NCOL
      BUFF(IC, IR, IL)=0.
      5 CONTINUE
C
C4-----FOR EACH RIVER REACH ACCUMULATE RIVER FLOW (STEPS 5-15)
      10 DO 100 L=1, NRIVER
C
C5-----GET LAYER, ROW & COLUMN OF CELL CONTAINING REACH.
      IL=RIVR(1, L)
      IR=RIVR(2, L)
      IC=RIVR(3, L)
C
C6-----IF CELL IS EXTERNAL MOVE ON TO NEXT REACH.
      IF(IBOUND(IC, IR, IL).LE.0)GO TO 100
C
C7-----GET RIVER PARAMETERS FROM RIVER LIST.
      HRIV=RIVR(4, L)
      CRIV=RIVR(5, L)
      RBOT=RIVR(6, L)
      HHNEW=HNEW(IC, IR, IL)

```

```

C
C8-----COMPARE HEAD IN AQUIFER TO BOTTOM OF RIVERBED.
C
C9-----AQUIFER HEAD > BOTTOM THEN RATE=CRIV*(HRIV-HNEW).
      IF(HHNEW.GT.RBOT)RATE=CRIV*(HRIV-HHNEW)
C
C10-----AQUIFER HEAD < BOTTOM THEN RATE=CRIV*(HRIV-RBOT)
      IF(HHNEW.LE.RBOT)RATE=CRIV*(HRIV-RBOT)
C
C11-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IRIVCB<0).
      IF(IRIVCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
1      KPER,KSTP,L,IL,IR,IC,RATE
900 FORMAT(1H0,4A4,' PERIOD',I3,' STEP',I3,' REACH',I4,
1      ' LAYER',I3,' ROW',I4,' COL',I4,' RATE',G15.7)
C
C12-----IF C-B-C FLOW TERMS ARE TO BE SAVED THEN ADD RATE TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+RATE
C
C13-----SEE IF FLOW IS INTO AQUIFER OR INTO RIVER.
      IF(RATE)94,100,96
C
C14-----AQUIFER IS DISCHARGING TO RIVER SUBTRACT RATE FROM RATOUT.
94 RATOUT=RATOUT-RATE
      GO TO 100
C
C15-----AQUIFER IS RECHARGED FROM RIVER ADD RATE TO RATIN.
96 RATIN=RATIN+RATE
100 CONTINUE
C
C16-----IF C-B-C FLOW TERMS WILL BE SAVED CALL UBUDSV TO RECORD THEM.
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IRIVCB,BUFF,NCOL,NROW,
1      NLAY,IOUT)
C
C17-----MOVE RATES,VOLUMES & LABELS INTO ARRAYS FOR PRINTING.
200 VBVL(3,MSUM)=RATIN
      VBVL(4,MSUM)=RATOUT
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C18-----INCREMENT BUDGET TERM COUNTER
      MSUM=MSUM+1
C
C19-----RETURN
      RETURN
      END

```

List of Variables for RIV1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
CRIV	Module	Conductance of the bed of the river reach.
DELT	Global	Length of the current time step.
HHNEW	Module	HNEW (J,I,K), Single precision.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HRIV	Module	Head in the river.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
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 either printed or recorded (depending on IRIVCB) for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IRIVCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, river leakage for each reach will be printed whenever ICBCFL is set.
KPER	Global	Stress period counter.

List of Variables for Module RIV1BD (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for river reaches.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXRIVR	Package	Maximum number of river reaches active at any one time.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRIVER	Package	Number of river reaches active during the current stress period.
NROW	Global	Number of rows in the grid.
RATE	Module	Flow from the river into the cell. (Reverse the sign to get the flow into the river.)
RATIN	Module	Accumulator for the total flow into the flow field from rivers.
RATOUT	Module	Accumulator for the total flow out of flow field into rivers.
RBOT	Module	Elevation of the bottom of the riverbed.
RIVR	Package	DIMENSION (6,MXRIVR), For each reach: layer, row, column, riverhead, riverbed conductance, and elevation of the bottom of the riverbed.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the 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.

## CHAPTER 7

### RECHARGE PACKAGE

#### Conceptualization and Implementation

Infiltration from precipitation generally occurs evenly over a large area. Hence, it is called "areally distributed recharge." It is expressed in terms of flow rate per unit area which reduces to units of length per unit time such as cm/sec or in/hr. The volumetric rate of flow into a cell is the infiltration rate times the horizontal area of the cell. In equation form

$$QRCH_{i,j,k} = I_{i,j,k} * DELR_j * DELC_i \quad (67)$$

where  $I_{i,j,k}$  is the infiltration rate ( $Lt^{-1}$ ). Notice that the recharge rate is independent of the head in the cell.

The recharge rate is stored in a two-dimensional array (RECH) with one element for each horizontal cell location. The layer to which the recharge is applied can be specified using one of three options (NRCHOP):

NRCHOP = 1, recharge only affects the uppermost layer;

NRCHOP = 2, recharge at each horizontal location affects the layer specified in an indicator array (IRCH) specified by the user; and

NRCHOP = 3, recharge affects the uppermost active cell in each vertical column.

The recharge rate is read as volumetric flow per unit area. That rate is multiplied by the horizontal cell area to get the volumetric flow rate. If

option 2 is specified, the indicator array (IRCH) is also read. The indicator array contains, for each horizontal cell location, the layer number to which recharge is applied for that horizontal location.

The finite-difference equation for cell  $i,j,k$  is

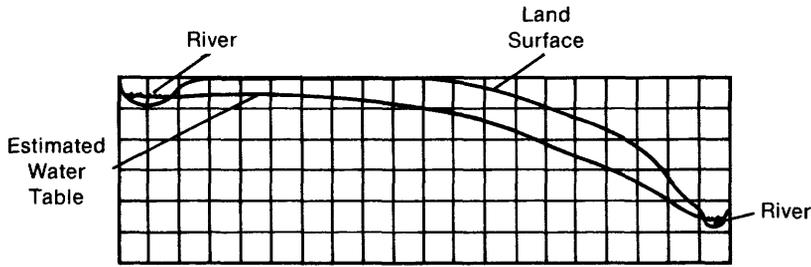
$$\begin{aligned}
 & CV_{i,j,k-1/2}h_{i,j,k-1} + CC_{i-1/2,j,k}h_{i-1,j,k} + CR_{i,j-1/2,k}h_{i,j-1,k} \\
 & + ( - CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
 & - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h_{i,j,k} + CR_{i,j+1/2,k}h_{i,j+1,k} \\
 & + CC_{i+1/2,j,k}h_{i+1,j,k} + CV_{i,j,k+1/2}h_{i,j,k+1} = RHS_{i,j,k} \qquad (68)
 \end{aligned}$$

where

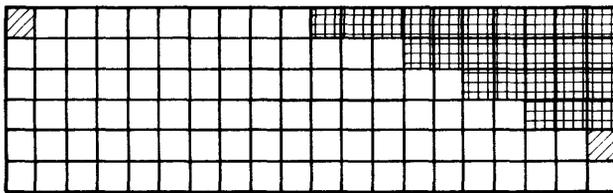
$RHS_{i,j,k}$  is the sum of all terms independent of head at the end of the time step ( $L^3t^{-1}$ ); and

$HCOF_{i,j,k}$  is the sum of all coefficients of head at the end of the time step other than conductances between cells ( $L^2t^{-1}$ ).

During the formulation phase of each iteration, the recharge rate is added to the accumulator in which RHS is formulated for the appropriate cell at each horizontal location (fig. 38). If option 1 is specified, the appropriate cell is in the top layer of the grid (layer 1). If option 2 is specified, the appropriate cell is the layer specified by the user in the indicator array (IRCH). If option 3 is specified, the appropriate cell is the uppermost active cell at the horizontal location which is not below a constant-head cell. If the uppermost active cell is below a constant-head cell, recharge is not applied to any cell because this recharge is assumed to be intercepted by the boundary.

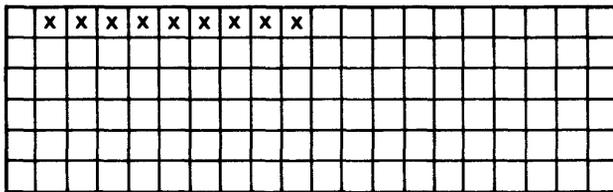


Vertical Cross-Section Showing Field Situation With Finite Difference Grid Superimposed



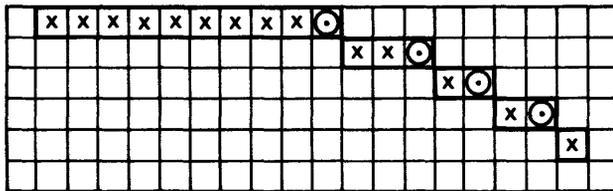
Status of Cells at End of Simulation

- Variable Head
- Constant Head
- Inactive



Cells Which Receive Recharge Under Option 1

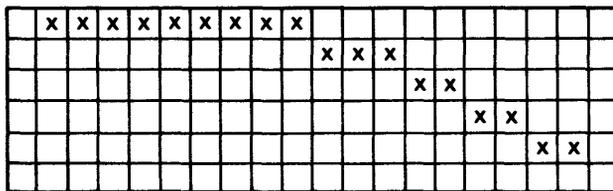
- Cell Which Receives Recharge



Cells Which Receive Recharge Under Option 2

- Cell Which Receives Recharge
- Inactive Cell Specified by User to Receive Recharge

Heavy Line Encloses Cells User Thought Would Receive Recharge Based on Estimated Water Table



Cells Which Receive Recharge Under Option 3

- Cell Which Receives Recharge

Figure 38.—Hypothetical problem showing which cells receive recharge under the three options available in the Recharge Package.

## Recharge Package Input

Input to the Recharge (RCH) Package is read from the unit specified in IUNIT(8).

FOR EACH SIMULATION

### RCH1AL

1. Data: NRCHOP    IRCHCB  
Format: I10       I10

FOR EACH STRESS PERIOD

### RCH1RP

2. Data: INRECH    INIRCH  
Format: I10       I10
3. Data: RECH(NCOL,NROW)  
Module: U2DREL

IF THE RECHARGE OPTION IS EQUAL TO 2

4. Data: IRCH(NCOL,NROW)  
Module: U2DINT

### Explanation of Fields Used in Input Instructions

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

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

IRCHCB--is a flag and a unit number.

If  $IRCHCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If  $IRCHCB \leq 0$ , cell-by-cell flow terms will not be printed or recorded.

INRECH--is the RECH read flag.

If  $INRECH \geq 0$ , an array of recharge rates, (RECH) is read.

If  $INRECH < 0$ , recharge rates from the preceding stress period are used.

INIRCH--is the IRCH read flag. When NRCHOP is two,

If  $INIRCH \geq 0$ , an array of layer numbers (IRCH) is read.

If  $INIRCH < 0$ , the array (IRCH) used in the preceding stress period is reused.

Note: When NRCHOP is one or three, INIRCH is ignored.

RECH--is the recharge rate. Read only if INRECH is greater than or equal to zero.

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

SAMPLE INPUT TO THE RECHARGE PACKAGE USING RECHARGE OPTION 1

DATA ITEM	EXPLANATION	INPUT RECORDS
-----------	-------------	---------------

1	{NRCHOP, IRCHGB} _____	1	0
2	Stress period 1---{INRECH} _____	12	3.17E-8 (10F4.0)
3	Control record for recharge array _____		

Recharge rates

1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.1
1.0	1.0	1.0	1.0	1.1	1.1
1.0	1.0	1.0	1.1	1.1	1.1
1.0	1.0	1.1	1.1	1.1	1.1
1.1	1.1	1.1	1.1	1.1	1.1

2	Stress period 2---{INRECH} _____	-1	
2	Stress period 3---{INRECH} _____	-1	
3	Stress period 4---{INRECH} _____	1	
	Control record for recharge array _____	12	3.17E-8 (10F4.0)

Recharge rates

1.2	1.2	1.2	1.2	1.2	1.3
1.2	1.2	1.2	1.3	1.4	1.4
1.2	1.2	1.3	1.4	1.4	1.4
1.0	1.0	1.0	1.1	1.1	1.1
1.2	1.3	1.3	1.4	1.4	1.4
1.3	1.3	1.4	1.4	1.4	1.4

SAMPLE INPUT TO THE RECHARGE PACKAGE USING RECHARGE OPTION 2

DATA ITEM	EXPLANATION	INPUT RECORDS
-----------	-------------	---------------

1	{NRCHOP, IRCHGB} _____	2	0
2	Stress period 1---{INRECH, INIRCH} _____	1	1
3	Control record for recharge array _____	0	3.17E-8
4	Control record for layer indicator array _____	12	1 (2012)

Layer numbers

1	2	2	2	3
1	2	2	2	2
1	1	2	2	2
1	1	1	1	2
1	1	1	1	1
1	1	1	1	1

2	Stress period 2---{INRECH, INIRCH} _____	1	-1
3	Control record for recharge array _____	0	1.56E-8
2	Stress period 3---{INRECH, INIRCH} _____	-1	-1

FIELDS IN ARRAY CONTROL RECORDS ARE---{ LOCAT, CONST, FMTIN, IPRN }

## Module Documentation for the Recharge Package

The Recharge Package (RCH1) consists of four modules, all of which are called by the MAIN program. The modules are:

- RCH1AL     Allocates space to contain recharge rate (RECH) and, if option 2 is specified, the layer-indicator array (IRCH).
  
- RCH1RP     Reads recharge rates (in flow per unit area) and indicator array (if option 2 is specified). Multiplies recharge rate by cell area.
  
- RCH1FM     Adds the inverse of the recharge rate to the accumulator in which RHS is formulated.
  
- RCH1BD     Calculates the rate and accumulated volume of recharge into the flow system.

### Narrative for Module RCH1AL

This module allocates space in the X array to store data relating to areally distributed recharge.

1. Print a message identifying the package.
2. Read and print the option indicator (NRCHOP) and the unit number for cell-by-cell flow terms (IRCHCB).
3. See if the recharge option (NRCHOP) is legal. If NRCHOP is illegal (not 1, 2, or 3), print a message saying the option is illegal. Do not allocate storage. STOP.
4. If NRCHOP is legal, print NRCHOP.
5. If cell-by-cell flow terms are to be recorded, print the unit number where they will be recorded.
6. Allocate space for the recharge array (RECH). Space is allocated by setting the first element of RECH (LCRECH) equal to the location (ISUM) of the first unused element in the X array and adding the size of the array to ISUM.
7. If the recharge option (NRCHOP) is equal to two, allocate space for a layer-indicator array (IRCH).
8. Calculate and print the number of elements in the X array used by the Recharge Package.
9. RETURN.

## Flow Chart for Module RCH1AL

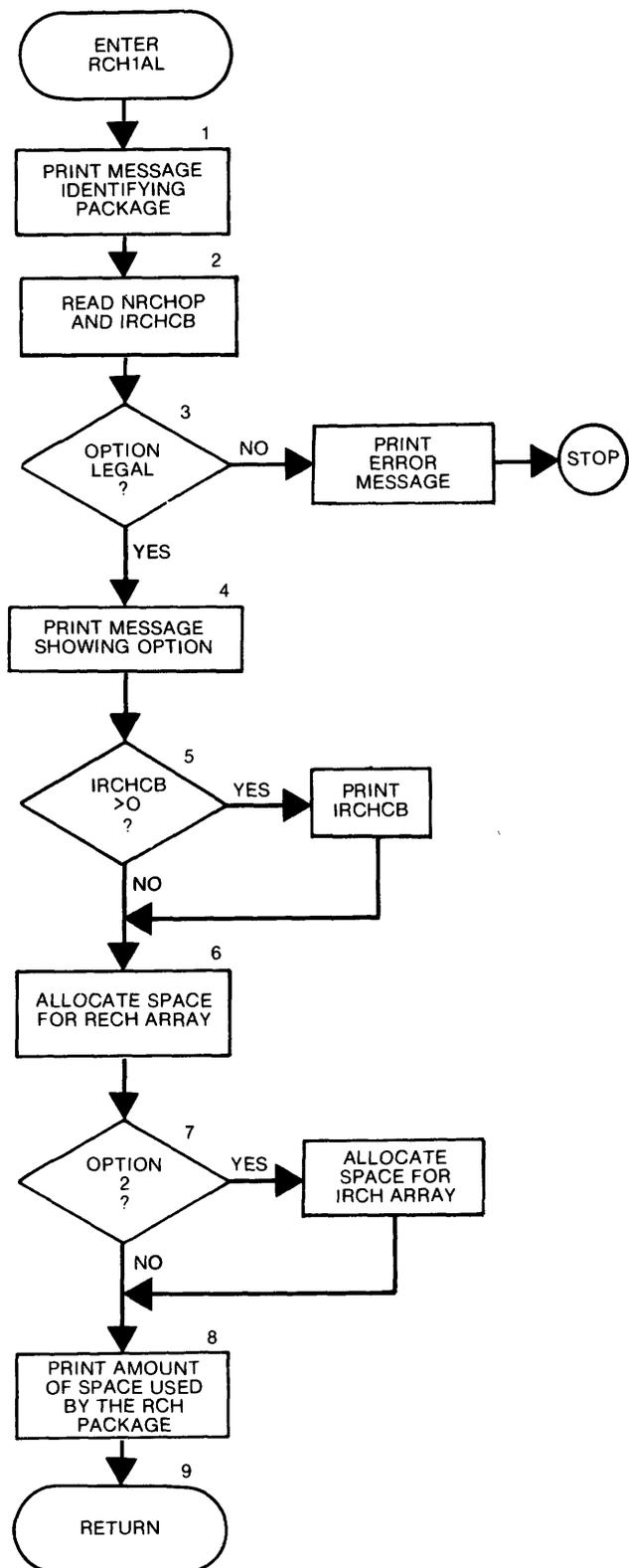
NRCHOP is the recharge option.

- 1 - Recharge is to the top layer.
- 2 - Recharge is to the layer specified by the user in the indicator array (IRCH).
- 3 - Recharge is to the upper-most active cell.

IRCHCB is the unit number on which cell-by-cell flow terms for recharge will be written.

RECH is an array which contains a recharge rate for each horizontal cell location.

IRCH is an array which contains the layer number to which recharge is applied for each horizontal location. It is used only if option 2 has been specified.



```

          SUBROUTINE RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
C          NCOL,NROW,IN,IOUT,IRCHCB)
C
C-----VERSION 0939 08DEC1983 RCH1AL
C *****
C ALLOCATE ARRAY STORAGE FOR RECHARGE
C *****
C
C          SPECIFICATIONS:
C -----
C -----
C
C1-----IDENTIFY PACKAGE.
          WRITE(IOUT,1)IN
          1 FORMAT(1H0,'RCH1 -- RECHARGE PACKAGE, VERSION 1, 12/08/83',
          2' INPUT READ FROM UNIT',I3)
C
C2-----READ NRCHOP AND IRCHCB.
          READ(IN,2)NRCHOP,IRCHCB
          2 FORMAT(2I10)
C
C3-----CHECK TO SEE THAT OPTION IS LEGAL.
          IF(NRCHOP.GE.1.AND.NRCHOP.LE.3)GO TO 200
C
C3A-----IF ILLEGAL PRINT A MESSAGE AND ABORT SIMULATION
          WRITE(IOUT,8)
          8 FORMAT(1X,'ILLEGAL OPTION CODE. SIMULATION ABORTING')
          STOP
C
C4-----IF OPTION IS LEGAL PRINT OPTION CODE.
          200 IRK=ISUM
             IF(NRCHOP.EQ.1) WRITE(IOUT,201)
             201 FORMAT(1X,'OPTION 1 -- RECHARGE TO TOP LAYER')
             IF(NRCHOP.EQ.2) WRITE(IOUT,202)
             202 FORMAT(1X,'OPTION 2 -- RECHARGE TO ONE SPECIFIED NODE IN EACH',
             1 ' VERTICAL COLUMN')
             IF(NRCHOP.EQ.3) WRITE(IOUT,203)
             203 FORMAT(1X,'OPTION 3 -- RECHARGE TO HIGHEST ACTIVE NODE IN EACH',
             1 ' VERTICAL COLUMN')
C
C5-----IF CELL-BY-CELL FLOW TERMS TO BE SAVED THEN PRINT UNIT #
          IF(IRCHCB.GT.0) WRITE(IOUT,204) IRCHCB
          204 FORMAT(1X,'CELL-BY-CELL FLOW TERMS WILL BE RECORDED ON UNIT',I3)
C
C6-----ALLOCATE SPACE FOR THE RECHARGE ARRAY(RECH).
          LCRECH=ISUM
          ISUM=ISUM+NCOL*NROW
C
C7-----IF OPTION 2 THEN ALLOCATE SPACE FOR INDICATOR ARRAY(IRCH)
          IF(NRCHOP.NE.2)GO TO 300
          LCIRCH=ISUM
          ISUM=ISUM+NCOL*NROW
C
C8-----CALCULATE AND PRINT AMOUNT OF SPACE USED BY RECHARGE.
          300 IRK=ISUM-IRK
             WRITE(IOUT,4)IRK
             4 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED FOR RECHARGE')
             ISUM1=ISUM-1
             WRITE(IOUT,5)ISUM1,LENX
             5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
             IF(ISUM1.GT.LENX)WRITE(IOUT,6)
             6 FORMAT(1X,' ***X ARRAY MUST BE MADE LARGER***')
C
C9-----RETURN
          RETURN
          END

```

List of Variables for Module RCH1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IN	Package	Primary unit number from which input for this package will be read.
IOUT IRCHCB	Global Package	Primary unit number for all printed output. IOUT = 6. Flag. IRCHCB $\leq$ 0, cell-by-cell flow terms will not be printed or recorded. IRCHCB $>$ 0 and ICBCFL $\neq$ 0, cell-by-cell flow terms for the RCH1 Package will be recorded on UNIT = IRCHCB.
IRK	Module	Before this module allocates space, IRK is set equal to ISUM. After allocation, IRK is subtracted from ISUM to get the amount of space in the X array allocated by this module.
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 LCIRCH	Module Package	ISUM-1. Location in the X array of the first element of array IRCH.
LCRECH	Package	Location in the X array of the first element of array RECH.
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 NRCHOP	Global Package	Number of columns in the grid. Recharge option: = 1, recharge is to the top grid layer. = 2, recharge is to the grid layer specified in array IRCH. = 3, recharge is to the highest variable-head cell which is not below a constant-head cell.
NROW	Global	Number of rows in the grid.

### Narrative for Module RCH1RP

This module reads data used to calculate the terms which represent areally distributed recharge.

1. Read the values INRECH and INIRCH which indicate whether the data contained in arrays RECH and IRCH used during the last stress period are to be used for the current stress period.

2. Test INRECH to see where the recharge rate (RECH) is coming from. If INRECH is less than zero, the recharge rate used in the last stress period will be used again in this stress period. Print a message to that effect. GO TO STEP 5.

3. If INRECH is greater than or equal to zero, CALL U2DREL to read the recharge rate (RECH).

4. Multiply the specified recharge rates by the cell areas to get the volumetric-recharge rate.

5. If the recharge option (NRCHOP) is not equal to two, a layer-indicator array is not needed. GO TO STEP 8.

6. If INIRCH is less than zero, the data in IRCH left over from the last stress period will be used in this stress period. Print a message to that effect. GO TO STEP 8.

7. If INIRCH is greater than or equal to zero, CALL U2DINT to read the IRCH array.

8. RETURN.

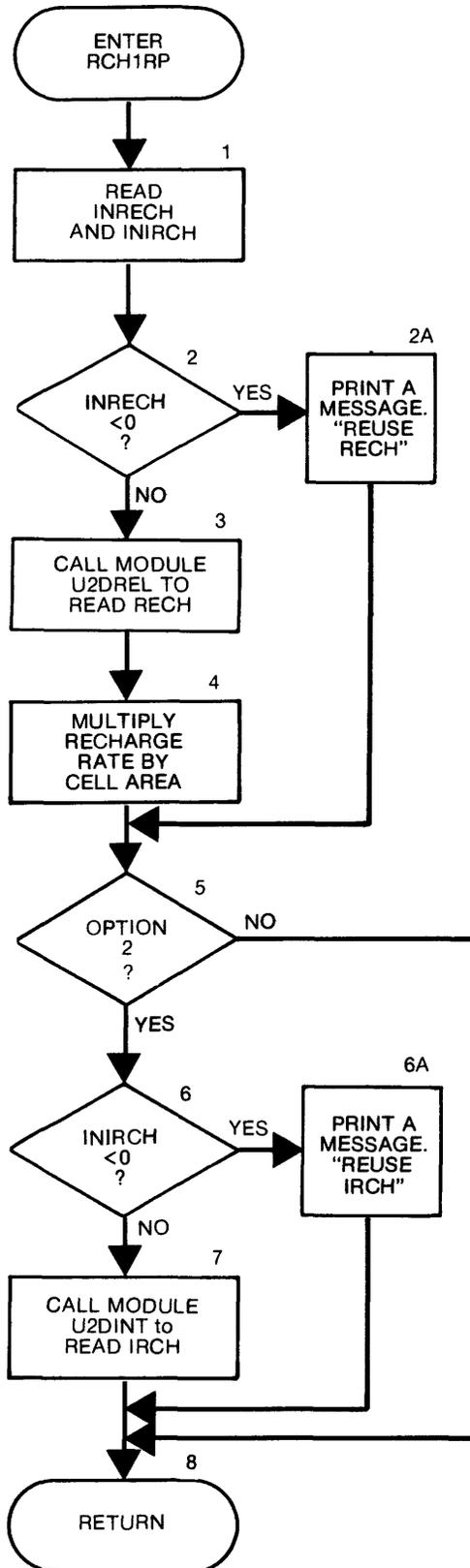
## Flow Chart for Module RCH1RP

INRECH is a flag which, when set, indicates that recharge rates (RECH) should be read for the current stress period. If it is clear ( $< 0$ ), recharge rates from the last stress period should be reused.

INIRCH is a flag similar to INRECH used for the layer indicator array IRCH.

RECH is an array containing a recharge rate for every horizontal cell location.

IRCH is an array containing a recharge indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location which gets recharge. It is used only if the recharge option (NRCHOP) is equal to two.



```

SUBROUTINE RCH1RP(NRCHOP,IRCH,RECH,DELR,DELC,NROW,NCOL,
C              NLAY,IN,IOUT)

```

```

C
C-----VERSION 1513 22DEC1982 RCH1RP
C *****
C READ RECHARGE RATES
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
1 ANAME(6,2),DELR(NCOL),DELC(NROW)
C
C DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
1 ANAME(6,1) /' ','RECH','ARGE',' LAY','ER I','NDEX'/
C DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
1 ANAME(6,2) /' ',' ',' ',' ',' ','RECH','ARGE'/
C -----
C
C1-----READ FLAGS SHOWING WHETHER DATA IS TO BE REUSED.
READ(IN,4)INRECH,INIRCH
4 FORMAT(2I10)
C
C2-----TEST INRECH TO SEE WHERE RECH IS COMING FROM.
IF(INRECH.GE.0)GO TO 32
C
C2A-----IF INRECH<0 THEN REUSE RECHARGE ARRAY FROM LAST STRESS PERIOD
WRITE(IOUT,3)
3 FORMAT(1H0,'REUSING RECH FROM LAST STRESS PERIOD')
GO TO 55
C
C3-----IF INRECH=>0 THEN CALL U2DREL TO READ RECHARGE RATE.
32 CALL U2DREL(RECH,ANAME(1,2),NROW,NCOL,0,IN,IOUT)
C
C4-----MULTIPLY RECHARGE RATE BY CELL AREA TO GET VOLUMETRIC RATE.
DO 50 IR=1,NROW
DO 50 IC=1,NCOL
RECH(IC,IR)=RECH(IC,IR)*DELR(IC)*DELC(IR)
50 CONTINUE
C
C5-----IF NRCHOP=2 THEN A LAYER INDICATOR ARRAY IS NEEDED.
55 IF (NRCHOP.NE.2)GO TO 60
C
C6-----IF INIRCH<0 THEN REUSE LAYER INDICATOR ARRAY.
IF(INIRCH.GE.0)GO TO 58
WRITE(IOUT,2)
2 FORMAT(1H0,'REUSING IRCH FROM LAST STRESS PERIOD')
GO TO 60
C
C7-----IF INIRCH=>0 CALL U2DINT TO READ LAYER IND ARRAY(IRCH)
58 CALL U2DINT(IRCH,ANAME(1,1),NROW,NCOL,0,IN,IOUT)
C
C8-----RETURN
60 RETURN
END

```

List of Variables for Module RCH1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ANAME	Module	Label for printout of the input array.
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
IC	Module	Index for columns.
IN	Package	Primary unit number from which input for this package will be read.
INIRCH	Module	Flag. ≥ 0, IRCH array will be read.  < 0, IRCH array already in memory from the last stress period will be used.
INRECH	Module	Flag. ≥ 0, RECH array will be read.  < 0, RECH array already in memory from the last stress period will be used.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IRCH	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal cell location to which recharge will be applied if the recharge option (NRCHOP) is equal to 2.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRCHOP	Package	Recharge option: = 1, recharge is to the top grid layer. = 2, recharge is to the grid layer specified in array IRCH. = 3, recharge is to the highest variable-head cell which is not below a constant-head cell.
NROW	Global	Number of rows in the grid.
RECH	Package	DIMENSION (NCOL,NROW), Recharge rate.

## Narrative for Module RCH1FM

This module adds terms representing areally distributed recharge to the accumulators in which the terms HCOF and RHS are formulated.

1. If the recharge option (NRCHOP) is equal to one, recharge is to the top layer. For each horizontal location, DO STEPS (a) AND (b).

(a) If the cell is external ( $IBOUND(I,J,K) \leq 0$ ), ignore it. SKIP STEP (b).

(b) Subtract the recharge rate from the RHS accumulator.

2. If the recharge option is two, recharge is only to the cells specified in the layer-indicator array (IRCH).

(a) Get the layer index from the layer-indicator array (IRCH).

(b) If the cell is external, ignore it. SKIP STEP (c).

(c) Subtract the recharge rate from the RHS accumulator.

3. If the recharge option is three, recharge is in the uppermost internal cell. For each horizontal cell location:

(a) If the cell is constant head, there will be no recharge below it. Move on to the next horizontal cell location.

(b) If the cell is no flow, move down a cell and go back to (a).

(c) Subtract the recharge rate from the RHS accumulator. Move on to the next horizontal cell location.

4. RETURN

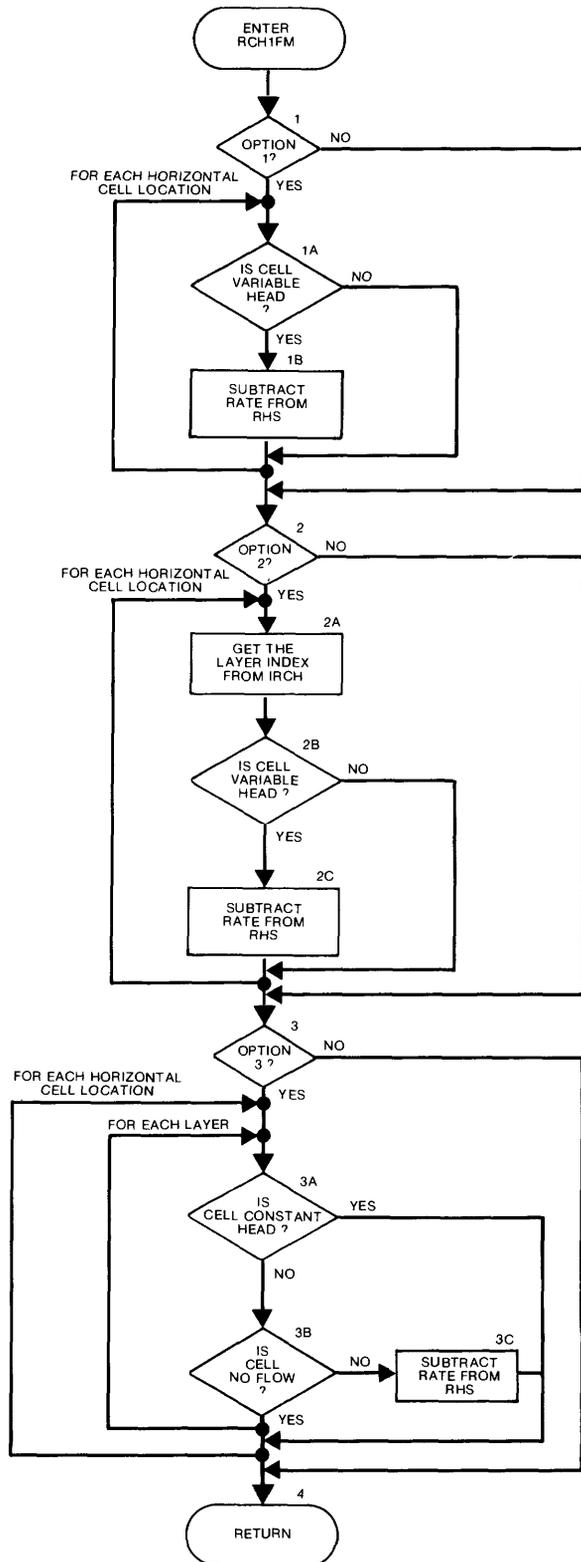
## Flow Chart for Module RCH1FM

RHS is the right hand side of the finite-difference equation. It includes all terms that are independent of head at the end of the time step.

IRCH is an array which contains the layer number to which recharge is applied for each horizontal location. It is used only if option 2 has been specified.

NRCHOP is the recharge option.

- 1 - Recharge is to the top layer.
- 2 - Recharge is to the layer specified by the user in the indicator array (IRCH).
- 3 - Recharge is to the uppermost active cell.



```

      SUBROUTINE RCH1FM(NRCHOP,IRCH,RECH,RHS,IBOUND,NCOL,
1                      NROW,NLAY,IOUT)
C
C-----VERSION 1518 22DEC1982 RCH1FM
C*****
C  ADD RECHARGE TO RHS
C*****
C
C      SPECIFICATIONS:
C-----
C  DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C-----
C1-----IF NRCHOP IS 1 RECHARGE IS IN TOP LAYER. LAYER INDEX IS 1.
      IF(NRCHOP.NE.1) GO TO 15
C
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C1A-----IF CELL IS EXTERNAL THERE IS NO RECHARGE INTO IT.
      IF(IBOUND(IC,IR,1).LE.0)GO TO 10
C
C1B-----SUBTRACT RECHARGE RATE FROM RIGHT-HAND-SIDE.
      RHS(IC,IR,1)=RHS(IC,IR,1)-RECH(IC,IR)
      10 CONTINUE
      GO TO 100
C
C2-----IF OPTION IS 2 THEN RECHARGE IS INTO LAYER IN INDICATOR ARRAY
      15 IF(NRCHOP.NE.2)GO TO 25
      DO 20 IR=1,NROW
      DO 20 IC=1,NCOL
C
C2A-----LAYER INDEX IS IN INDICATOR ARRAY.
      IL=IRCH(IC,IR)
C
C2B-----IF THE CELL IS EXTERNAL THERE IS NO RECHARGE INTO IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 20
C
C2C-----SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-RECH(IC,IR)
      20 CONTINUE
      GO TO 100
C
C3-----IF OPTION IS 3 RECHARGE IS INTO HIGHEST INTERNAL CELL.
      25 IF(NRCHOP.NE.3)GO TO 100
C      CANNOT PASS THROUGH CONSTANT HEAD NODE
      DO 30 IR=1,NROW
      DO 30 IC=1,NCOL
      DO 28 IL=1,NLAY
C
C3A-----IF CELL IS CONSTANT HEAD MOVE ON TO NEXT HORIZONTAL LOCATION.
      IF(IBOUND(IC,IR,IL).LT.0) GO TO 30
C
C3B-----IF CELL IS INACTIVE MOVE DOWN A LAYER.
      IF (IBOUND(IC,IR,IL).EQ.0)GO TO 28
C
C3C-----SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-RECH(IC,IR)
      GO TO 30
      28 CONTINUE
      30 CONTINUE
      100 CONTINUE
C
C4-----RETURN
      RETURN
      END

```

List of Variables for Module RCH1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IRCH	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal cell location to which recharge will be applied if the recharge option (NRCHOP) is equal to 2.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRCHOP	Package	Recharge option: = 1, recharge is to the top grid layer. = 2, recharge is to the grid layer specified in array IRCH. = 3, recharge is to the highest variable-head cell which is not below a constant-head cell.
NROW	Global	Number of rows in the grid.
RECH	Package	DIMENSION (NCOL,NROW), Recharge rate.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

## Narrative for Module RCH1BD

This module calculates rates and volumes added to the aquifer by areally distributed recharge.

1. Clear the rate accumulators RATIN and RATOUT.
2. If cell-by-cell flow terms will be saved, clear the buffer (BUFF) in which they will be accumulated.
3. If the recharge option is one, the recharge goes into the top layer. Process the horizontal locations one at a time.
  - (a) If the cell is external, do not calculate budget.
  - (b) If cell-by-cell flow terms will be saved, add recharge to the buffer.
  - (c) If the recharge is positive, add it to RATIN; otherwise, add it to RATOUT.
4. If the recharge option is two, recharge goes into the layer specified in indicator array (IRCH). Process the horizontal locations one at a time.
  - (a) Get the cell layer from indicator array (IRCH).
  - (b) If the cell is external, do not calculate budget.
  - (c) If cell-by-cell flow terms will be saved, add the recharge to the buffer.
  - (d) If the recharge is positive, add it to RATIN; otherwise, add it to RATOUT.

5. If the recharge option is three, the recharge goes into the top variable-head cell provided there is not a constant-head cell above it. Process the horizontal locations one at a time. Start with the top cell and work down.

(a) If the cell is inactive, there is no recharge into that cell; move down to the next one.

(b) If the cell is constant, there is no recharge at this horizontal location; move on to the next horizontal location.

(c) If cell-by-cell flow terms are to be saved, add the recharge to the buffer.

(d) If the recharge is positive, add it to RATIN; otherwise, add it to RATOUT.

6. If cell-by-cell flow terms will be saved, call module UBUDSV to write the buffer (BUFF) onto disk.

7. Move RATIN and RATOUT into the VBVL array for printing by BAS10T.

8. Add RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS10T.

9. Move the recharge budget-term labels to VBNM for printing by BAS10T.

10. Increment the budget-term counter (MSUM).

11. RETURN.

## Flow Chart for Module RCH1BD

RATIN is an accumulator to which all flows into the aquifer are added.

RATOUT is an accumulator to which all flows out of the aquifer are added.

BUFFER is an array in which values are stored as they are being gathered for printing or recording.

NRCHOP is the recharge option.

1 - Recharge is to the top layer.

2 - Recharge is to the layer specified by the user in the indicator array (IRCH).

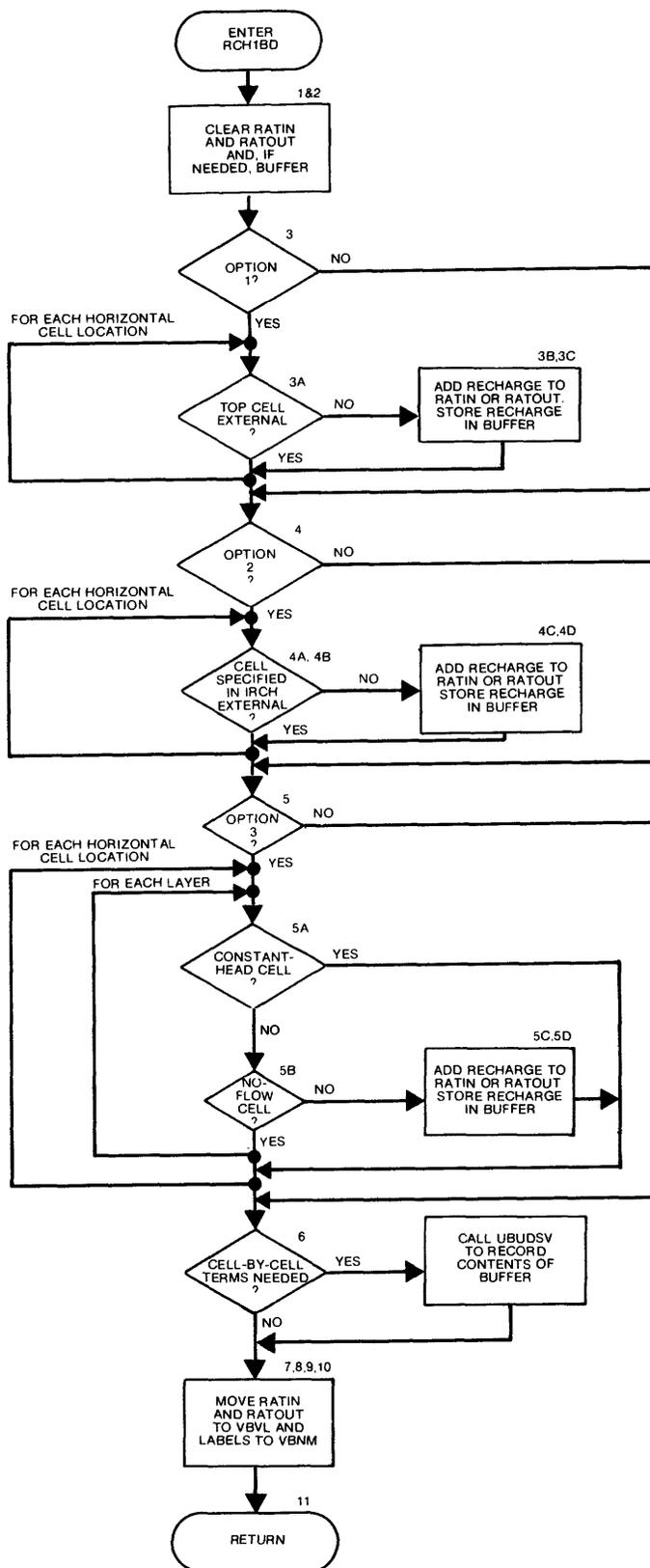
3 - Recharge is to the uppermost active cell.

IRCH is an array containing a recharge indicator for each horizontal cell. It is used only if the recharge option (NRCHOP) is equal to two.

VBVL is a table of budget entries calculated by component-of-flow packages for use in calculating the volumetric budget.

VBNM is a table of labels for budget terms.

EXTERNAL: a cell is external if it is either no flow (inactive) or constant head.



```

SUBROUTINE RCH1BD(NRCHOP,IRCH,RECH,IBOUND,NROW,NCOL,NLAY,
1  DELT,VBVL,VBNM,MSUM,KSTP,KPER,IRCHCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1533 22DEC1982 RCH1BD
C *****
C CALCULATE VOLUMETRIC BUDGET FOR RECHARGE
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
1 IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY),
2 VBVL(4,20),VBNM(4,20)
C DIMENSION TEXT(4)
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' ',' ','RECH','ARGE'/
C -----
C
C1-----CLEAR THE RATE ACCUMULATORS.
RATIN=0.
RATOUT=0.
C
C2-----IF CELL-BY-CELL FLOW TERMS WILL BE SAVED THEN CLEAR THE BUFFER.
IBD=0
IF(ICBCFL.EQ.0 .OR. IRCHCB.LE.0) GO TO 5
IBD=1
DO 2 IL=1,NLAY
DO 2 IR=1,NROW
DO 2 IC=1,NCOL
BUFF(IC,IR,IL)=0.
2 CONTINUE
C
C3-----IF NRCHOP=1 RECH GOES INTO LAYER 1. PROCESS EACH HORIZONTAL
C3-----CELL LOCATION.
5 IF(NRCHOP.NE.1) GO TO 15
C
C ---RECHARGE IS APPLIED TO TOP LAYER
DO 10 IR=1,NROW
DO 10 IC=1,NCOL
C
C3A-----IF CELL IS EXTERNAL THEN DO NOT DO BUDGET FOR IT.
IF(IBOUND(IC,IR,1).LE.0)GO TO 10
Q=RECH(IC,IR)
C
C3B-----IF CELL-BY-CELL FLOW TERMS WILL BE SAVED THEN ADD RECH TO BUFF
IF(IBD.EQ.1) BUFF(IC,IR,1)=Q
C
C3C-----IF RECH POSITIVE ADD IT TO RATIN ELSE ADD IT TO RATOUT.
IF(Q) 8,10,7
7 RATIN=RATIN+Q
GO TO 10
8 RATOUT=RATOUT-Q
10 CONTINUE
GO TO 100
C
C4-----IF NRCHOP=2 RECH IS IN LAYER SHOWN IN INDICATOR ARRAY(IRCH).
C4-----PROCESS HORIZONTAL CELL LOCATIONS ONE AT A TIME.
15 IF(NRCHOP.NE.2)GO TO 25
DO 20 IR=1,NROW
DO 20 IC=1,NCOL
C
C4A-----GET LAYER INDEX FROM INDICATOR ARRAY(IRCH).
IL=IRCH(IC,IR)
C
C4B-----IF CELL IS EXTERNAL DO NOT CALCULATE BUDGET FOR IT.
IF(IBOUND(IC,IR,IL).LE.0)GO TO 20
Q=RECH(IC,IR)

```

```

C
C4C-----IF C-B-C FLOW TERMS WILL BE SAVED THEN ADD RECHARGE TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=Q
C
C4D-----IF RECHARGE IS POSITIVE ADD TO RATIN ELSE ADD IT TO RATOUT.
      IF(Q) 18,20,17
      17 RATIN=RATIN+Q
      GO TO 20
      18 RATOUT=RATOUT-Q
      20 CONTINUE
      GO TO 100
C
C5-----IF OPTION=3 RECHARGE IS INTO HIGHEST INTERNAL CELL. IT WILL NOT
C5-----PASS THROUGH A CONSTANT HEAD CELL. PROCESS HORIZONTAL CELL
C5-----LOCATIONS ONE AT A TIME.
      25 IF(NRCHOP.NE.3)GO TO 100
      DO 30 IR=1,NROW
      DO 30 IC=1,NCOL
      DO 28 IL=1,NLAY
C
C5A-----IF CELL IS CONSTANT HEAD MOVE ON TO NEXT HORIZONTAL LOCATION.
      IF(IBOUND(IC,IR,IL).LT.0) GO TO 30
C
C5B-----IF CELL IS INACTIVE MOVE DOWN TO NEXT CELL.
      IF (IBOUND(IC,IR,IL).EQ.0)GO TO 28
      Q=RECH(IC,IR)
C
C5C-----IF C-B-C FLOW TERMS TO BE SAVED THEN ADD RECHARGE TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=Q
C
C5D-----IF RECH IS POSITIVE ADD IT TO RATIN ELSE ADD IT TO RATOUT.
      IF(Q) 27,30,26
      26 RATIN=RATIN+Q
      GO TO 30
      27 RATOUT=RATOUT-Q
      GO TO 30
      28 CONTINUE
      30 CONTINUE
C
      100 CONTINUE
C
C6-----IF C-B-C FLOW TERMS TO BE SAVED CALL UBUDSV TO WRITE THEM.
      IF (IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IRCHCB,BUFF,NCOL,NROW,
1                                NLAY,IOUT)
C
C7-----MOVE TOTAL RECHARGE RATE INTO VBVL FOR PRINTING BY BAS10T.
      VBVL(4,MSUM)=RATOUT
      VBVL(3,MSUM)=RATIN
C
C8-----ADD RECHARGE FOR TIME STEP TO RECHARGE ACCUMULATOR IN VBVL.
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
C
C9-----MOVE BUDGET TERM LABELS TO VBNM FOR PRINT BY MODULE BAS_0T.
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C10-----INCREMENT BUDGET TERM COUNTER.
      MSUM=MSUM+1
C
C11-----RETURN
      RETURN
      END

```

List of Variables for Module RCH1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
DELT	Global	Length of the current time step.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
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 for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IRCH	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal cell location to which recharge will be applied if the recharge option (NRCHOP) is equal to 2.
IRCHCB	Package	Flag. IRCHCB ≤ 0, cell-by-cell flow terms will not be recorded or printed.  IRCHCB > 0 and ICBCFL ≠ 0, cell-by-cell flow terms for the RCH1 Package will be recorded on UNIT = IRCHCB.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.

List of Variables for Module RCH1BD (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
NLAY	Global	Number of layers in the grid.
NRCHOP	Package	Recharge option: = 1, recharge is to the top grid layer. = 2, recharge is to the grid layer specified in array IRCH. = 3, recharge is to the highest variable-head cell which is not below a constant-head cell.
NROW	Global	Number of rows in the grid.
Q	Module	Flow from recharge into a cell. (Reverse the sign to get flow out of the cell.)
RATIN	Module	Accumulator for the total flow into the flow field from recharge.
RATOUT	Module	Accumulator for the total flow out of the flow field to recharge.
RECH	Package	DIMENSION (NCOL,NROW), Recharge rate.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the 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.

## CHAPTER 8

### WELL PACKAGE

#### Conceptualization and Implementation

A recharging well can be viewed as a source of water which is not affected by the head in the aquifer. A discharging well is a recharging well with a negative recharge rate. For the sake of this discussion, the well will be assumed to be screened in only a single cell. Thus for each cell containing a well, the recharge rate must be added to the right side of the finite-difference equation.

A list containing the location and rate for each well is maintained. The list contains four values for each entry: row, column, and layer of the cell, and the rate at which the well recharges the aquifer. At each iteration, for each variable-head cell  $i,j,k$  containing a pumping well, the well rate is added to the accumulator in which  $RHS_{i,j,k}$  is formulated.

## Well Package Input

Input for the Well (WEL) Package is read from the unit specified in IUNIT(2).

FOR EACH SIMULATION

WEL1AL

1. Data:   MXWELL   IWELCB  
Format: I10     I10

FOR EACH STRESS PERIOD

WEL1RP

2. Data:   ITMP  
Format: I10
3. Data:   Layer   Row     Column   Q  
Format: I10    I10     I10     F10.0

(Input item 3 normally consists of one record for each well.  
If ITMP is negative or zero, item 3 is not read.)

### Explanation of Fields Used in Input Instructions

MXWELL--is the maximum number of wells used at any time.

IWELCB--is a flag and a unit number.

If IWELCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If IWELCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IWELCB < 0, well recharge will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

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

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

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

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



## Module Documentation for the Well Package

The Well Package (WELL) consists of four modules, all of which are called by the MAIN program. The modules are:

- WELL1AL      Allocates space for the list of wells (WELL).
  
- WELL1RP      Reads location and recharge rate for all wells.  
                 Note: Discharge rate is entered as a negative  
                 number.
  
- WELL1FM      Subtracts the recharge rate from the term RHS for  
                 each cell containing pumping wells.
  
- WELL1BD      Calculates the rates and accumulated volume of  
                 recharge to or discharge from the flow system  
                 by pumping wells.

## Narrative for Module WEL1AL

This module allocates space in the X array to store the list of wells. The X array is a pool of memory space from which space is allocated for tables, lists, and arrays.

1. Print a message identifying the package and initialize NWELLS (a counter containing the number of wells).

2. Read and print MXWELL (the maximum number of wells) and IWELBD (the unit number for cell-by-cell flow terms or a flag indicating that cell-by-cell flow terms should be printed).

3. Set LCWELL, which will point to the first element in the well list (WELL), equal to ISUM, which is currently pointing to the first unallocated element in the X array.

4. Calculate the amount of space needed for the well list (four values for each cell--row, column, layer, and rate) and add it to ISUM.

5. Print the number of elements in the X array used by the Well Package.

6. If the pointer to the lowest unallocated element in the X array (ISUM) is greater than the length of the X array (LENX), print a message warning that the X array will have to be enlarged.

7. RETURN.

## Flow Chart for Module WEL1AL

MXWELL is the maximum number of wells that will be active at any one time during the simulation.

IWELCB is a flag and a unit number.

If  $IWELCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If  $IWELCB = 0$ , cell-by-cell flow terms will not be printed or recorded.

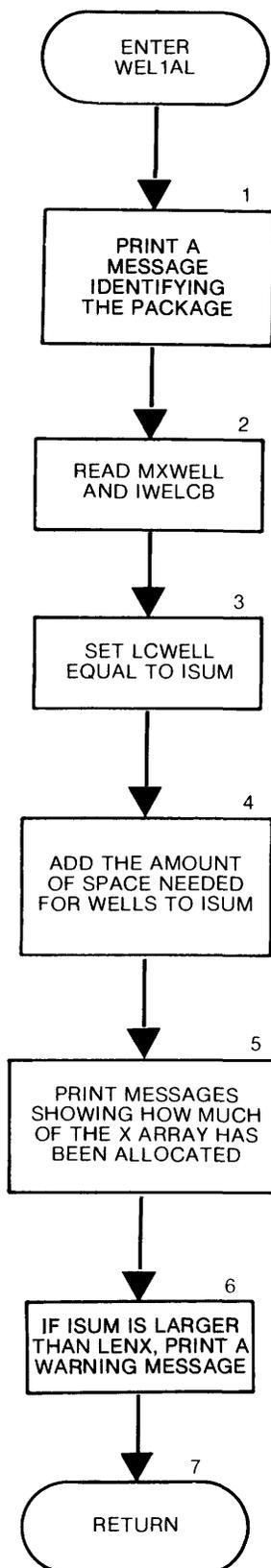
If  $IWELCB < 0$ , well recharge will be printed whenever ICBCFL is set.

LCWELL is a location pointer to the first storage location occupied by the well list.

ISUM is the location of the lowest unallocated storage location in the X array.

X array is the pool of memory space allocated for storing specific tables, arrays, and lists.

LENX is the size of the X array.



```

SUBROUTINE WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,IN,IOUT,
1          IWELCB)
C
C-----VERSION 0933 08DEC1983 WEL1AL
C *****
C ALLOCATE ARRAY STORAGE FOR WELL PACKAGE
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----IDENTIFY PACKAGE AND INITIALIZE NWELLS
WRITE(IOUT,1)IN
1 FORMAT(1H0,'WEL1 -- WELL PACKAGE, VERSION 1, 12/08/83',
2' INPUT READ FROM',I3)
NWELLS=0
C
C2-----READ MAX NUMBER OF WELLS AND
C2-----UNIT OR FLAG FOR CELL-BY-CELL FLOW TERMS.
READ(IN,2) MXWELL,IWELCB
2 FORMAT(2I10)
WRITE(IOUT,3) MXWELL
3 FORMAT(1H , 'MAXIMUM OF',I5, ' WELLS')
IF(IWELCB.GT.0) WRITE(IOUT,9) IWELCB
9 FORMAT(1X, 'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
IF(IWELCB.LT.0) WRITE(IOUT,8)
8 FORMAT(1X, 'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3-----SET LCWELL EQUAL TO LOCATION OF WELL LIST IN X ARRAY.
LCWELL=ISUM
C
C4-----ADD AMOUNT OF SPACE USED BY WELL LIST TO ISUM.
ISP=4*MXWELL
ISUM=ISUM+ISP
C
C5-----PRINT NUMBER OF SPACES IN X ARRAY USED BY WELL PACKAGE.
WRITE(IOUT,4) ISP
4 FORMAT(1X,I6, ' ELEMENTS IN X ARRAY ARE USED FOR WELLS')
ISUM1=ISUM-1
WRITE(IOUT,5) ISUM1,LENX
5 FORMAT(1X,I6, ' ELEMENTS OF X ARRAY USED OUT OF',I7)
C
C6-----IF THERE ISN'T ENOUGH SPACE IN THE X ARRAY THEN PRINT
C6-----A WARNING MESSAGE.
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X, ' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C7-----RETURN
RETURN
END

```

List of Variables for Module WEL1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISP	Module	Number of words in the X array allocated by this module.
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	ISUM-1.
IWELCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, well recharge will be printed whenever ICBCFL is set.
LCWELL	Package	Location in the X array of the first element of array WELL.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXWELL	Package	Maximum number of wells active at any one time.
NWELLS	Package	Number of wells active during the current stress period.

### Narrative for Module WEL1RP

This module reads data to build the WELL list.

1. Read ITMP.

(a) If ITMP is less than zero, the well data read for the last stress period will be reused. Print a message to that effect and RETURN.

(b) If ITMP is greater than or equal to zero, it is equal to the number of wells (NWELLS) in the current stress period.

2. If the number of wells (NWELLS) in the current stress period is greater than the number specified as the maximum for the simulation (MXWELL), STOP.

3. Print the number of wells in the current stress period (NWELLS).

4. If there are no wells in the current stress period (NWELLS), bypass further well processing.

5. For each well, read and print the layer, row, column, and well recharge rate.

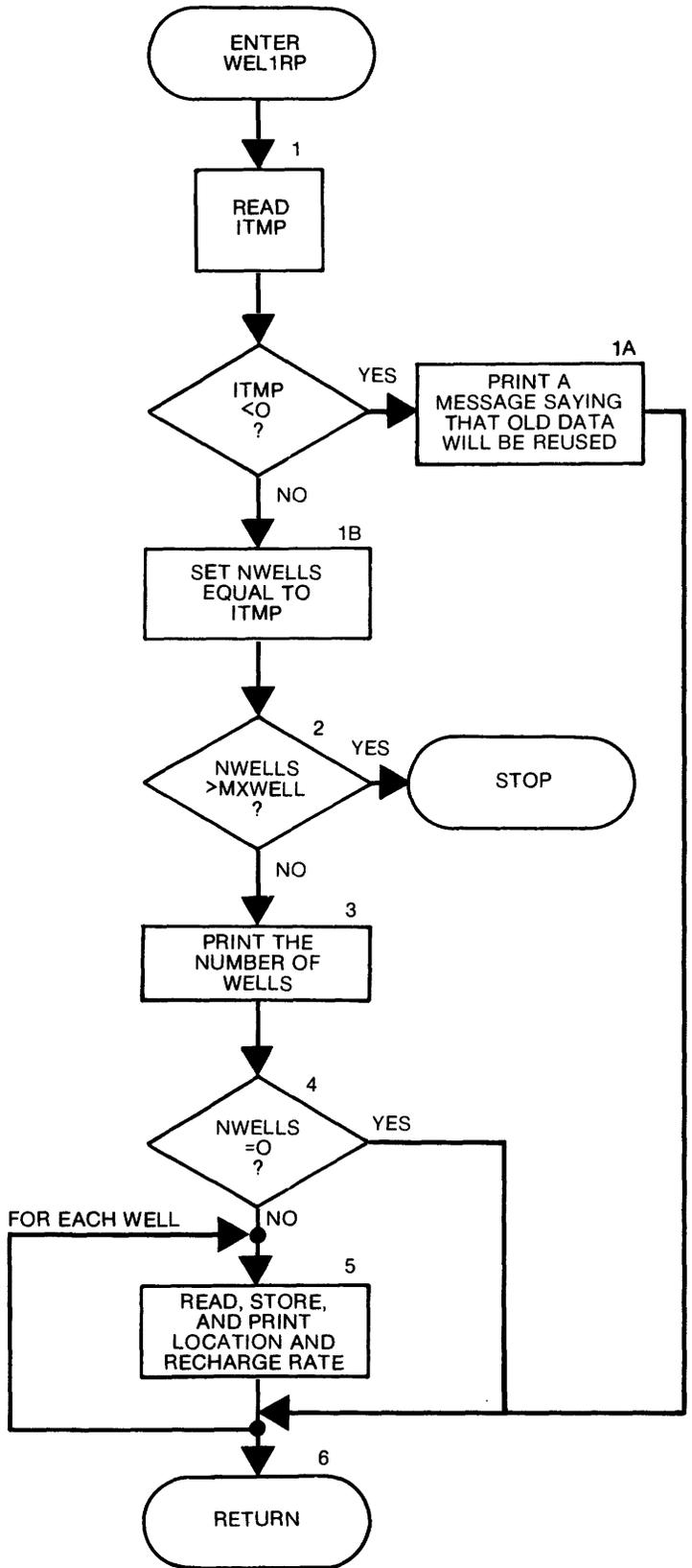
6. RETURN.

Flow Chart for Module WEL1RP

ITMP is a flag and/or the number of wells. If it is less than zero, it is a flag which indicates that the well data from the last stress period will be reused. If it is greater than or equal to zero, it is the number of wells active during the current stress period.

NWELLS is the number of wells active during the current stress period.

MXWELL is the maximum number of wells which will be active at any one time during the simulation.



```

SUBROUTINE WELIRP(WELL,NWELLS,MXWELL,IN,IOUT)
C
C
C-----VERSION 1544 22DEC1982 WELIRP
C *****
C READ NEW WELL LOCATIONS AND STRESS RATES
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION WELL(4,MXWELL)
C -----
C
C1-----READ ITMP(NUMBER OF WELLS OR FLAG SAYING REUSE WELL DATA)
      READ (IN,1) ITMP
      1 FORMAT(I10)
      IF(ITMP.GE.0) GO TO 50
C
C1A-----IF ITMP LESS THAN ZERO REUSE DATA. PRINT MESSAGE AND RETURN.
      WRITE(IOUT,6)
      6 FORMAT(1H0,'REUSING WELLS FROM LAST STRESS PERIOD')
      RETURN
C
C1B-----ITMP=>0. SET NWELLS EQUAL TO ITMP.
      50 NWELLS=ITMP
      IF(NWELLS.LE.MXWELL) GO TO 100
C
C2-----NWELLS>MXWELL. PRINT MESSAGE. STOP.
      WRITE(IOUT,99) NWELLS,MXWELL
      99 FORMAT(1H0,'NWELLS('',I4,'') IS GREATER THAN MXWELL('',I4,'')')
      STOP
C
C3-----PRINT NUMBER OF WELLS IN CURRENT STRESS PERIOD.
      100 WRITE (IOUT,2) NWELLS
      2 FORMAT(1H0,10X,I4,' WELLS')
C
C4-----IF THERE ARE NO ACTIVE WELLS IN THIS STRESS PERIOD THEN RETURN
      IF(NWELLS.EQ.0) GO TO 260
C
C5-----READ AND PRINT LAYER,ROW,COLUMN AND RECHARGE RATE.
      WRITE(IOUT,3)
      3 FORMAT(1H ,47X,'LAYER ROW COL STRESS RATE WELL NO.'/
      1,48X,45('-'))
      DO 250 II=1,NWELLS
      READ (IN,4) K,I,J,Q
      4 FORMAT(3I10,F10.0)
      WRITE (IOUT,5) K,I,J,Q,II
      5 FORMAT(48X,I3,I8,I7,G16.5,I8)
      WELL(1,II)=K
      WELL(2,II)=I
      WELL(3,II)=J
      WELL(4,II)=Q
      250 CONTINUE
C
C6-----RETURN
      260 RETURN
      END

```

List of Variables for Module WEL1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
I	Module	Row number of cell containing well.
II	Module	Index for wells.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of wells. ≥ 0, number of wells active during the current stress period. < 0, same wells active during the last stress period will be active during the current stress period.
J	Module	Column number of cell containing well.
K	Module	Layer number of cell containing well.
MXWELL	Package	Maximum number of wells active at any one time.
NWELLS	Package	Number of wells active during the current stress period.
Q	Module	Rate at which the well recharges the aquifer.
WELL	Package	DIMENSION (4,MXWELL), For each well: layer, row, column, and recharge rate of the well.

### Narrative for Module WEL1FM

This module adds terms representing well recharge to the accumulator in which the term RHS is formulated.

1. If NWELLS is less than or equal to zero in the current stress period, there are no wells. RETURN.

2. For each well in the WELL list:

(a) If the cell containing the well is external ( $IBOUND(IC,IR,IL) \leq 0$ ), bypass processing on this well and go on to the next well.

(b) If the cell containing the well is active, add the recharge rate to the accumulator RHS for that cell.

3. RETURN.

```

          SUBROUTINE WEL1FM(NWELLS,MXWELL,RHS,WELL,IBOUND,
1          NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1001 26AUG1982 WEL1FM
C
C          *****
C          ADD WELL FLOW TO SOURCE TERM
C          *****
C
C          SPECIFICATIONS:
C          -----
C          DIMENSION RHS(NCOL,NROW,NLAY),WELL(4,MXWELL),
1          IBOUND(NCOL,NROW,NLAY)
C          -----
C1-----IF NUMBER OF WELLS <= 0 THEN RETURN.
          IF(NWELLS.LE.0) RETURN
C
C2-----PROCESS EACH WELL IN THE WELL LIST.
          DO 100 L=1,NWELLS
            IR=WELL(2,L)
            IC=WELL(3,L)
            IL=WELL(1,L)
            Q=WELL(4,L)
C
C2A-----IF THE CELL IS INACTIVE THEN BYPASS PROCESSING.
            IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C2B-----IF THE CELL IS VARIABLE HEAD THEN ADD RECHARGE RATE
            TO THE RHS ACCUMULATOR.
            RHS(IC,IR,IL)=RHS(IC,IR,IL)-Q
          100 CONTINUE
C
C3-----RETURN
          RETURN
          END

```

List of Variables for Module WEL1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
L	Module	Index for wells.
MXWELL	Package	Maximum number of wells active at any one time.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NWELLS	Package	Number of wells active during the current stress period.
Q	Module	Rate at which the well recharges the aquifer.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.
WELL	Package	DIMENSION (4,MXWELL), For each well: layer, row, column, and recharge rate of the well.

## Narrative for Module WEL1BD

This module calculates rates and volumes transferred between the aquifer and wells.

1. Clear the rate accumulators RATIN and RATOUT and the flag (IBD) which indicates that cell-by-cell flow terms should be recorded on a disk.
2. If there are no wells, skip down to step 7.
3. Determine if the cell-by-cell flow terms for wells will be written on a disk. They will be if (1) this is the proper time step (ICBCFL is not equal to zero), (2) if the channel for well-budget terms (IWELCB) is greater than zero, and (3) if the number of wells (NWELLS) is greater than zero.
4. If budget terms are to be written on a disk, set IBD = 1 and clear the buffer (BUFF) in which they will be accumulated.
5. If the number of wells in the current stress period (NWELLS) is not equal to zero, then for each cell in the well list:
  - (a) If the cell containing the well is external ( $IBOUND(I,J,K) \leq 0$ ), bypass further processing of the cell.
  - (b) If the user has requested that cell-by-cell rates be printed ( $IWELCB < 0$  and  $ICBCFL \neq 0$ ), print the rate (Q).
  - (c) If the budget terms are to be saved on a disk, add the recharge rate (Q) to the buffer (BUFF).
  - (d) If Q is positive, add it to RATIN.
  - (e) If Q is negative, add it to RATOUT.
6. If the cell-by-cell flow terms are to be recorded, call module UBUDSV to write the contents of buffer (BUFF) onto the disk.
7. Move RATIN and RATOUT into the VBVL array for printing by BAS10T.
8. Add RATIN and RATOUT multiplied by the time-step length to the volume accumulators in the VBVL array for printing by BAS10T.
9. Move the well budget term labels to VBNM for printing by BAS10T.
10. Increment the budget-term counter (MSUM).
11. RETURN.

## Flow Chart for Module WEL1BD

RATIN is an accumulator to which all flows into the aquifer are added.

RATOUT is an accumulator to which all flows out of the aquifer are added.

IBD is a flag which, if set, causes cell-by-cell flow terms for river well flow to be recorded.

BUFFER is an array in which values are stored as they are being gathered for printing or recording.

EXTERNAL: a cell is said to be external if it is either no flow or constant head (i.e., an equation is not formulated for the cell).

RECHARGE is the rate at which the well recharges the aquifer. A discharging well is represented by a negative rate.

IWELCB is a flag and a unit number.

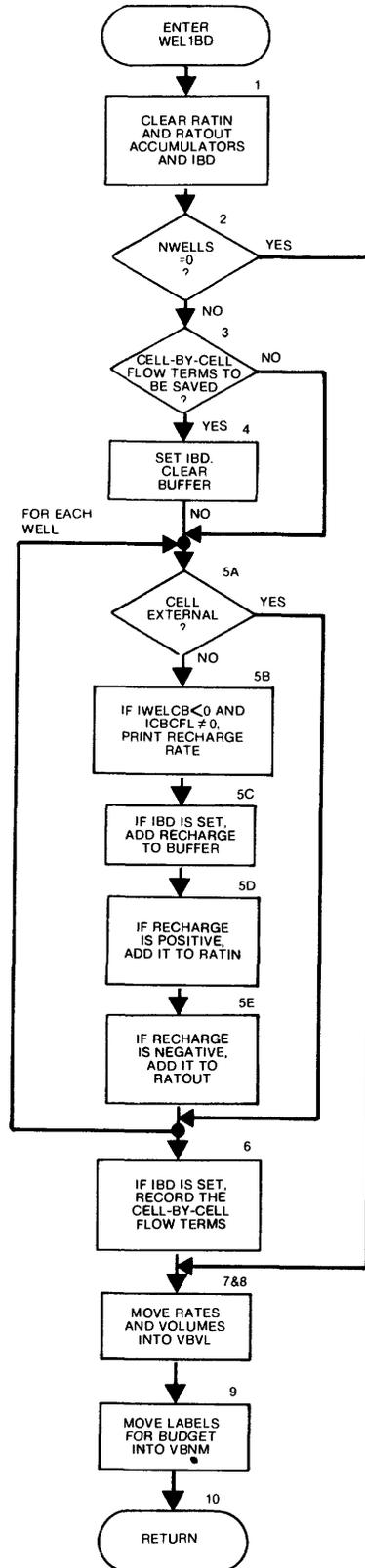
If  $IWELCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If  $IWELCB = 0$ , cell-by-cell flow terms will not be printed or recorded.

If  $IWELCB < 0$ , well recharge rate will be printed whenever ICBCFL is set.

ICBCFL is a flag.

If  $ICBCFL \neq 0$ , cell-by-cell flow terms will be either printed or recorded (depending on IWELCB) for the current time step.



```

SUBROUTINE WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,WELL,IBOUND,DELT,
1          NCOL,NROW,NLAY,KSTP,KPER,IWELCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1449 20MAY1983 WEL1BD
C *****
C CALCULATE VOLUMETRIC BUDGET FOR WELLS
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION VBNM(4,MSUM),VBVL(4,MSUM),WELL(4,MXWELL),
1          IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
C DIMENSION TEXT(4)
C
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'  ','  ','  ' W','ELLS'/
C -----
C
C1-----CLEAR RATIN AND RATOUT ACCUMULATORS.
C          RATIN=0.
C          RATOUT=0.
C          IBD=0
C
C2-----IF THERE ARE NO WELLS DO NOT ACCUMULATE FLOW
C          IF(NWELLS.EQ.0) GO TO 200
C
C3-----TEST TO SEE IF CELL-BY-CELL FLOW TERMS WILL BE RECORDED.
C          IF(ICBCFL.EQ.0 .OR. IWELCB.LE.0) GO TO 60
C
C4-----IF CELL-BY-CELL FLOWS WILL BE SAVED THEN CLEAR THE BUFFER.
C          IBD=1
C          DO 50 IL=1,NLAY
C          DO 50 IR=1,NROW
C          DO 50 IC=1,NCOL
C          BUFF(IC,IR,IL)=0.
C          50 CONTINUE
C
C5-----PROCESS WELLS ONE AT A TIME.
C          60 DO 100 L=1,NWELLS
C             IR=WELL(2,L)
C             IC=WELL(3,L)
C             IL=WELL(1,L)
C             Q=WELL(4,L)
C
C5A-----IF THE CELL IS EXTERNAL IGNORE IT.
C          IF(IBOUND(IC,IR,IL).LE.0)GO TO 100
C
C5B-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IWELCB<0).
C          IF(IWELCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
1          KPER,KSTP,L,IL,IR,IC,Q
C          900 FORMAT(1H0,4A4,' PERIOD',I3,' STEP',I3,' WELL',I4,
1          ' LAYER',I3,' ROW ',I4,' COL',I4,' RATE',G15.7)
C
C5C-----IF CELL-BY-CELL FLOWS ARE TO BE SAVED THEN ADD THEM TO BUFFER.

```

```

      IF (IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+Q
      IF (Q) 90,100,80
C
C5D-----PUMPING RATE IS POSITIVE(RECHARGE). ADD IT TO RATIN.
      80 RATIN=RATIN+Q
      GO TO 100
C
C5E-----PUMPING RATE IS NEGATIVE(DISCHARGE). ADD IT TO RATOUT.
      90 RATOUT=RATOUT-Q
      100 CONTINUE
C
C6-----IF CELL-BY-CELL FLOWS WILL BE SAVED CALL UBUDSV TO RECORD THEM
      IF (IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IWELCB,BUFF,NCOL,NROW,
      1                               NLAY,IOUT)
C
C7-----MOVE RATES INTO VBVL FOR PRINTING BY MODULE BAS10T.
      200 VBVL(3,MSUM)=RATIN
      VBVL(4,MSUM)=RATOUT
C
C8-----MOVE RATES TIMES TIME STEP LENGTH INTO VBVL ACCUMULATORS.
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
C
C9-----MOVE BUDGET TERM LABELS INTO VBNM FOR PRINTING.
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C10-----INCREMENT BUDGET TERM COUNTER(MSUM).
      MSUM=MSUM+1
C
C11-----RETURN
      RETURN
      END

```

List of Variables for Module WEL1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
DELT	Global	Length of the current time step.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
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 either printed or recorded (depending on IWELCB) for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IWELCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, well recharge rate will be printed whenever ICBCFL is set.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for wells.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXWELL	Package	Maximum number of wells active at any one time.

List of Variables for Module WEL1BD (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NWELLS	Package	Number of wells active during the current stress period.
Q	Module	Rate at which the well recharges the aquifer.
RATIN	Module	Accumulator for the total flow into the flow field from wells.
RATOUT	Module	Accumulator for the total flow out of the flow field into wells.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N) Rate for the current time step into the flow field. (2,N) Rate for the 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.
WELL	Package	DIMENSION (4,MXWELL), For each well: layer, row, column, and recharge rate of the well.

## CHAPTER 9

### DRAIN PACKAGE

#### Conceptualization and Implementation

The rate at which water seeps into a drain in the saturated zone of an aquifer is approximated in the model using the equation

$$QD_{i,j,k} = CD_{i,j,k}(h_{i,j,k} - d_{i,j,k}) = CD_{i,j,k}h_{i,j,k} - CD_{i,j,k}d_{i,j,k} \quad (69)$$

where

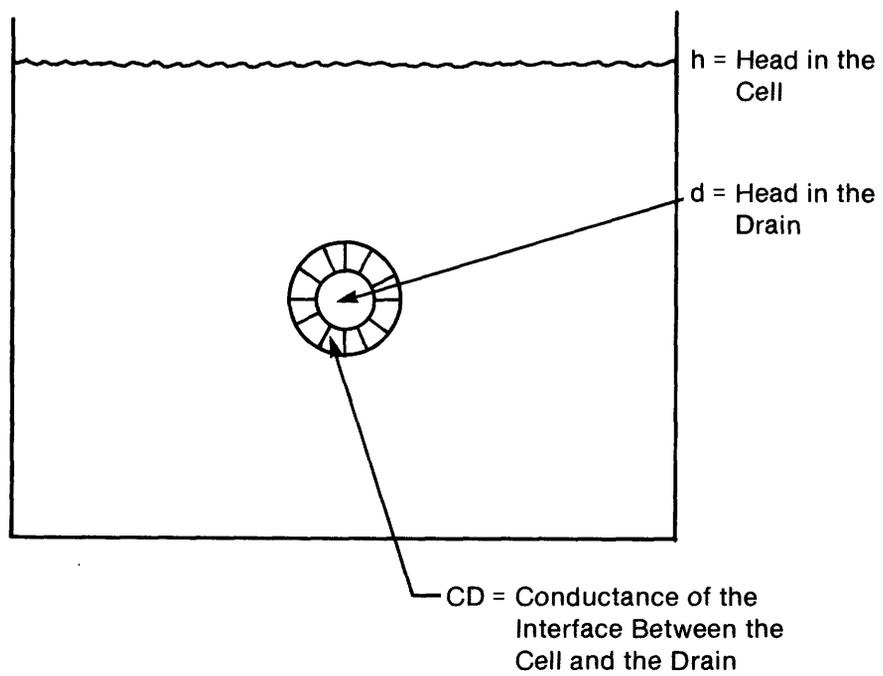
$QD_{i,j,k}$  is the rate water flows into the drain ( $L^3t^{-1}$ );

$d_{i,j,k}$  is the head in the drain (L);

$h_{i,j,k}$  is the head in the aquifer near the drain (L); and

$CD_{i,j,k}$  is the conductance of the interface between the aquifer and the drain ( $L^2t^{-1}$ ).

The coefficient  $CD_{i,j,k}$  (fig. 39) is the conductance of the interface between the drain and the porous material. It may be affected by size and frequency of openings in a drain tile, chemical precipitation around a tile, difference in permeability between the aquifer material and the backfill around a tile, and a low permeability bed in an open drain or the converging area of flow as the drain is approached (fig. 40). The head in the drain is assumed to be the elevation of the drain. Thus the flow into the drain is assumed to be proportional to the head above the drain (fig. 41). This equation only holds when the head in the aquifer is greater than the head in the drain. When the elevation of the drain is greater than the head in the aquifer, the flow into the drain,  $QD_{i,j,k}$ , is equal to zero.



$$Q = CD (h-d)$$

Figure 39.—Flow into a drain as a function of head in the aquifer and the elevation of the drain.

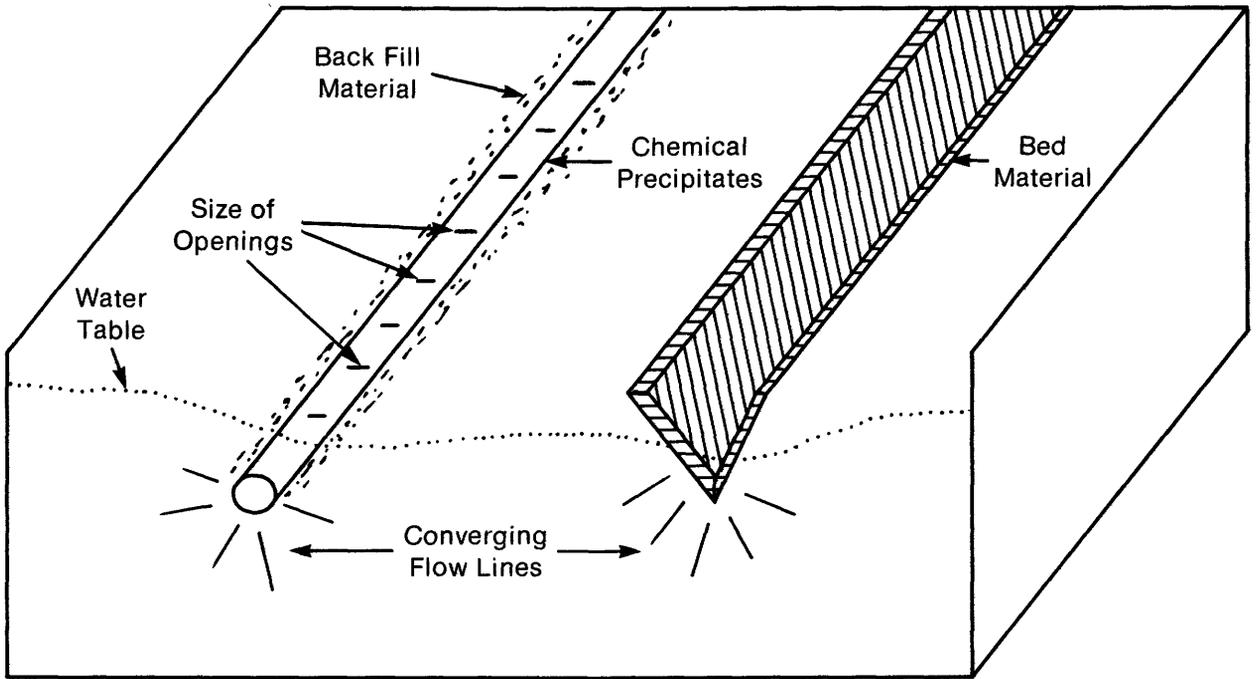
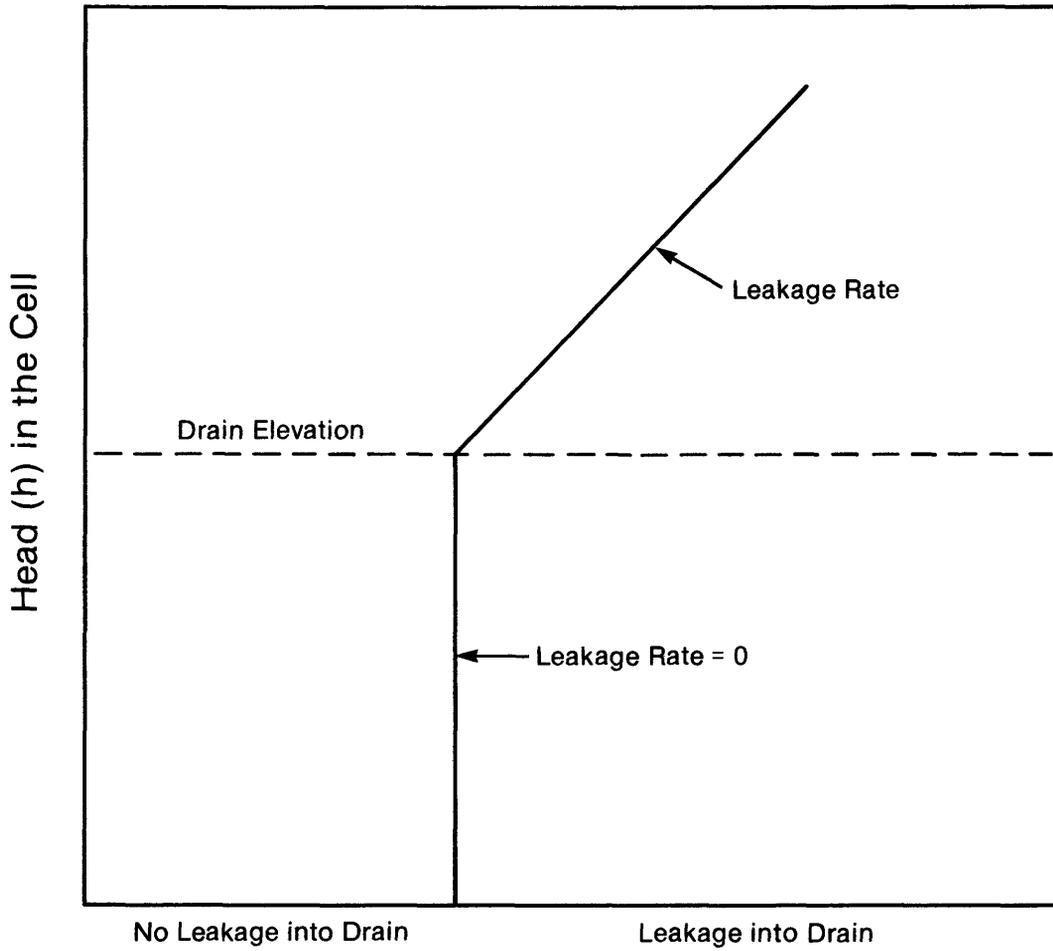


Figure 40.—Factors that may affect the conductance of the interface between an aquifer and a drain.



Leakage into a Drain

Figure 41.—Leakage into a drain as a function of head in the aquifer.

Data describing each drain is stored in a list. The contents of the list, which are specified by the user at the beginning of each stress period, consist of the row, column, layer, elevation, and conductance for each drain. During the formulation phase, the heads from the previous iteration are compared to the drain elevation. If the drain elevation is less than the head in the cell, the expressions  $-CD_{i,j,k}$  and  $-CD_{i,j,k}d_{i,j,k}$  are added to the accumulator  $HCOF_{i,j,k}$  and  $RHS_{i,j,k}$ , respectively. If the drain elevation is greater than or equal to the head in the cell, nothing is added to  $HCOF_{i,j,k}$  and  $RHS_{i,j,k}$ .

## Drain Package Input

Input to the Drain (DRN) Package is read from the unit specified in IUNIT(3).

FOR EACH SIMULATION

### DRN1AL

- |          |       |        |
|----------|-------|--------|
| 1. Data: | MXDRN | IDRNCB |
| Format:  | I10   | I10    |

FOR EACH STRESS PERIOD

### DRN1RP

- |          |       |     |     |           |       |
|----------|-------|-----|-----|-----------|-------|
| 2. Data: | ITMP  |     |     |           |       |
| Format:  | I10   |     |     |           |       |
| 3. Data: | Layer | Row | Col | Elevation | Cond  |
| Format:  | I10   | I10 | I10 | F10.0     | F10.0 |

(Input item 3 normally consists of one record for each drain.  
If ITMP is negative or zero, item 3 will not be read.)

### Explanation of Fields Used in Input Instructions

MXDRN--is the maximum number of drain cells active at one time.

IDRNCB--is a flag and a unit number.

If IDRNCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If IDRNCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IDRNCB < 0, drain leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

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

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

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

Elevation--is elevation of the drain.

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

SAMPLE INPUT TO THE DRAIN PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS	
1	{MXDRNR, IDRNCB}	3	
2	{ITMP} FOR FIRST STRESS PERIOD	3	
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR FIRST DRAIN	2	220.
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR SECOND DRAIN	2	225.
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR THIRD DRAIN	2	210.
2	{ITMP} FOR SECOND STRESS PERIOD	-1	
2	{ITMP} FOR THIRD STRESS PERIOD	-1	
2	{ITMP} FOR FOURTH STRESS PERIOD	2	
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR FIRST DRAIN	5	210.
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR SECOND DRAIN	6	220.
2	{ITMP} FOR FIFTH STRESS PERIOD	0	
2	{ITMP} FOR SIXTH STRESS PERIOD	-1	

## Module Documentation for the Drain Package

The Drain Package (DRN1) consists of four modules, all of which are called by the MAIN program. The modules are:

- DRN1AL      Allocates space for an array that contains  
                 the drain list (DRAI).
  
- DRN1RP      Reads location, drain elevation, and drain  
                 conductance of each cell containing a drain.
  
- DRN1FM      Adds the terms  $-CD_{i,j,k}$  and  $-CD_{i,j,k}d_{i,j,k}$  to  
                 the accumulators  $HCOF_{i,j,k}$  and  $RHS_{i,j,k}$ ,  
                 respectively.
  
- DRN1BD      Calculates the rates and accumulated volume of  
                 drainage from the flow system.

### Narrative for Module DRNIAL

This module allocates space in the X array to store the list of drains.

1. Print a message identifying the package and initialize NDRAIN (number of drains).
2. Read and print MXDRAN (the maximum number of drains) and IDRNCB (the file number for saving cell-by-cell flow terms or a flag indicating that cell-by-cell flow terms should be printed).
3. Set LCDRAI (which will point to the first element in the drain list) equal to ISUM (which points to the first unallocated element in the X array).
4. Calculate the amount of space needed for the drain list (five values for each drain--row, column, layer, drain elevation, and drain conductance).
5. Print the number of elements in the X array used by the Drain Package.
6. RETURN.

Flow Chart for Module DRN1AL

NDRAIN is the number of drains being simulated at any given time.

MXDRN is the maximum number of drains simulated.

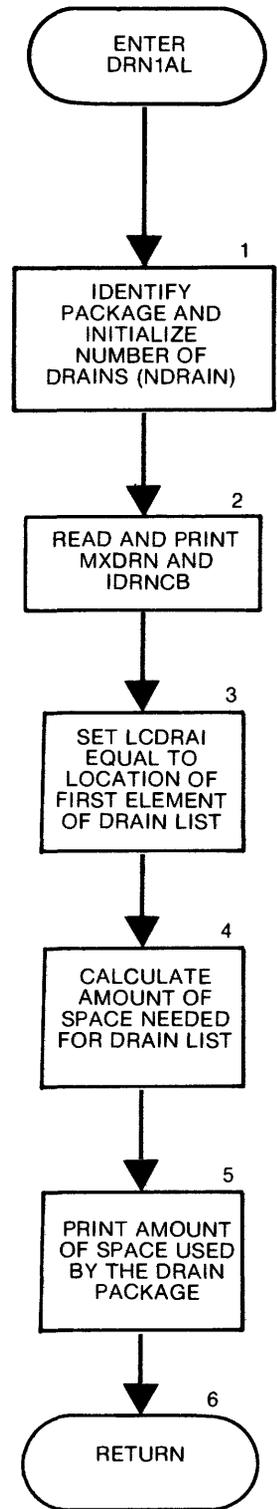
IDRNCB is a flag and a unit number.

If  $IDRNCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If  $IDRNCB = 0$ , cell-by-cell flow terms will not be printed or recorded.

If  $IDRNCB < 0$ , drain leakage for each drain will be printed whenever ICBCFL is set.

LCDRAI is the location, in the X array, of the list of drain data (DRAI).



```

SUBROUTINE DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,IN,IOUT,
1          IDRNCB)
C
C-----VERSION 0956 08DEC1983 DRN1AL
C *****
C ALLOCATE ARRAY STORAGE FOR DRAIN PACKAGE
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----IDENTIFY PACKAGE AND INITIALIZE NDRAIN.
WRITE(IOUT,1)IN
1 FORMAT(1H0,'DRN1 -- DRAIN PACKAGE, VERSION 1, 12/08/83',
2 ' INPUT READ FROM UNIT',I3)
NDRAIN=0
C
C2-----READ & PRINT MXDRN & IDRNCB(UNIT & FLAG FOR CELL-BY-CELL FLOW)
READ(IN,2) MXDRN,IDRNCB
2 FORMAT(2I10)
WRITE(IOUT,3) MXDRN
3 FORMAT(1H , 'MAXIMUM OF',I5, ' DRAINS')
IF(IDRNCB.GT.0) WRITE(IOUT,9) IDRNCB
9 FORMAT(1X, 'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
IF(IDRNCB.LT.0) WRITE(IOUT,8)
8 FORMAT(1X, 'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3-----SET LCDRAI EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
LCDRAI=ISUM
C
C4-----CALCULATE AMOUNT OF SPACE USED BY THE DRAIN PACKAGE.
ISP=5*MXDRN
ISUM=ISUM+ISP
C
C5-----PRINT AMOUNT OF SPACE USED BY DRAIN PACKAGE.
WRITE(IOUT,4) ISP
4 FORMAT(1X,I6, ' ELEMENTS IN X ARRAY ARE USED FOR DRAINS')
ISUM1=ISUM-1
WRITE(IOUT,5) ISUM1,LENX
5 FORMAT(1X,I6, ' ELEMENTS OF X ARRAY USED OUT OF',I7)
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X, ' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6-----RETURN
RETURN
END

```

List of Variables for Module DRN1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IDRNCB	Package	Flag and a unit number.  > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will be neither printed nor recorded. < 0, leakage for each drain will be printed.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISP	Module	Number of words in the X array allocated by this module.
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	ISUM - 1.
LCDRAI	Package	Location in the X array of the first element of array DRAI.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN Program.
MXDRN	Package	Maximum number of drains active at any one time.
NDRAIN	Package	Number of drains active during the current stress period.

### Narrative for Module DRN1RP

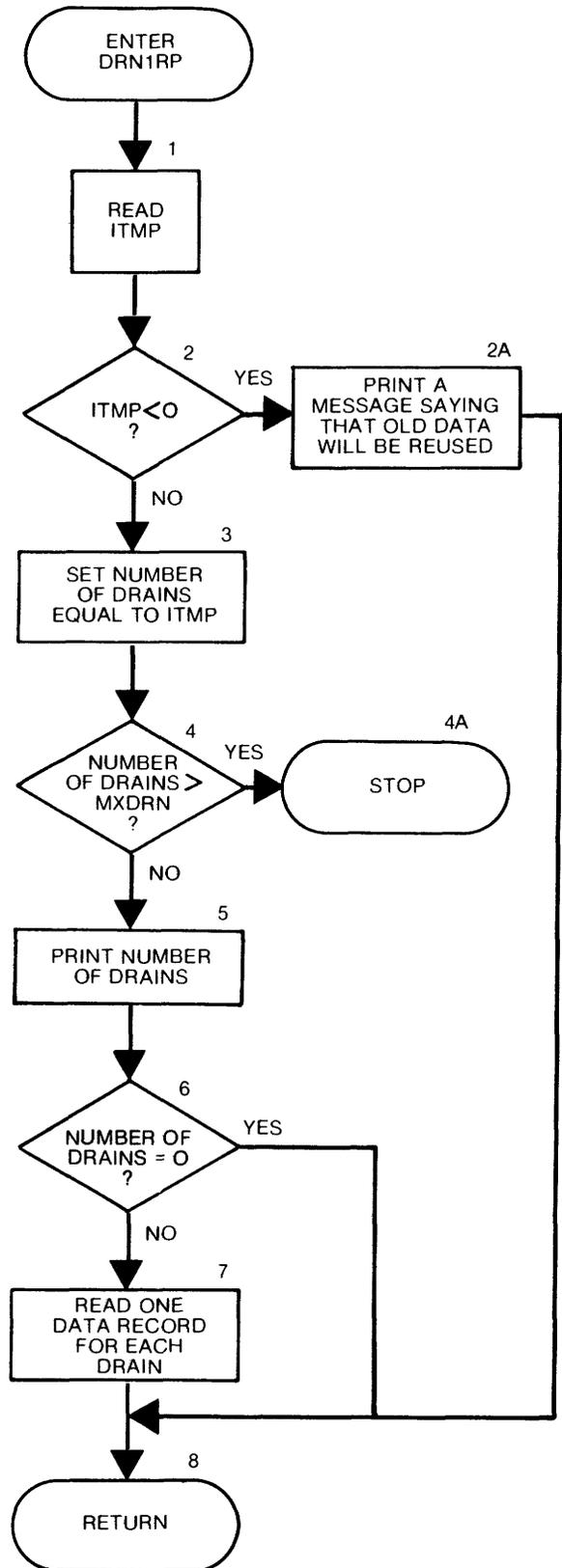
This module reads data to build the drain list.

1. Read ITMP. ITMP is the number of drains or a flag indicating that drain data from the previous stress period should be reused.
2. Test ITMP. If ITMP is less than zero, the drain data read for the last stress period will be reused. Print a message to that effect and RETURN.
3. If ITMP is greater than or equal to zero, it is the number of drains for this stress period. Set the number of drains (NDRAIN) in the current stress period equal to ITMP.
4. Compare the number of drains (NDRAIN) in the current stress period to the number specified as the maximum for the simulation (MXDRN). If NDRAIN is greater than MXDRN, STOP.
5. Print the number of drains in the current stress period (NDRAIN).
6. See if there are any drains. If there are no drains in the current stress period (NDRAIN = 0), bypass further drain processing.
7. Read and print the layer, row, column, elevation, and conductance for each drain.
8. RETURN.

Flow Chart for Module DRN1RP

ITMP is both a flag and a counter. If it is greater than or equal to zero, it is the number of drains to be simulated during the current stress period. If it is less than zero, it indicates that the drains simulated in the last stress period should be simulated in the current stress period.

MXDRN is the maximum number of drains to be simulated.



```

SUBROUTINE DRNIRP(DRAI,NDRAIN,MXDRN,IN,IOUT)
C
C
C-----VERSION 1603 25APR1983 DRNIRP
C *****
C READ DRAIN LOCATIONS, ELEVATIONS, AND CONDUCTANCES
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION DRAI(5,MXDRN)
C -----
C
C1-----READ ITMP (NUMBER OF DRAIN CELLS OR FLAG TO REUSE DATA)
      READ(IN,8) ITMP
      8 FORMAT(I10)
C
C2-----TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
      7 FORMAT(1H0,'REUSING DRAINS FROM LAST STRESS PERIOD')
      RETURN
C
C3-----IF ITMP=>0 THEN IT IS THE NUMBER OF DRAINS.
      50 NDRAIN=ITMP
      IF(NDRAIN.LE.MXDRN) GO TO 100
C
C4-----IF NDRAIN>MXDRN THEN STOP
      WRITE(IOUT,99) NDRAIN,MXDRN
      99 FORMAT(1H0,'NDRAIN(',I4,') IS GREATER THAN MXDRN(',I4,')')
      STOP
C
C5-----PRINT NUMBER OF DRAINS IN THIS STRESS PERIOD.
      100 WRITE(IOUT,1) NDRAIN
      1 FORMAT(1H0,'/1X,15,' DRAINS')
C
C6-----IF THERE ARE NO DRAINS THEN RETURN.
      IF(NDRAIN.EQ.0) GO TO 260
C
C7-----READ AND PRINT DATA FOR EACH DRAIN.
      WRITE(IOUT,3)
      3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X
      1,'COL ELEVATION CONDUCTANCE DRAIN NO. '/1X,15X,60('-'))
      DO 250 II=1,NDRAIN
      READ (IN,4) K,I,J,DRAI(4,II),DRAI(5,II)
      4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,DRAI(4,II),DRAI(5,II),II
      5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,I8)
      DRAI(1,II)=K
      DRAI(2,II)=I
      DRAI(3,II)=J
      250 CONTINUE
C
C8-----RETURN
      260 RETURN
C
      END

```

List of Variables for Module DRN1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
DRAI	Package	DIMENSION (5,MXDRN), For each drain: layer, row, column, head in drain, and conductance into drain.
I	Module	Index for rows.
II	Module	Index for drains.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of drains. $\geq 0$ , number of drains active during the current stress period. $< 0$ , same drains active during the last stress period will be active during the current stress period.
J	Module	Index for columns.
K	Module	Index for layers.
MXDRN	Package	Maximum number of drains active at any one time.
NDRAIN	Package	Number of drains active during the current stress period.

### Narrative for Module DRN1FM

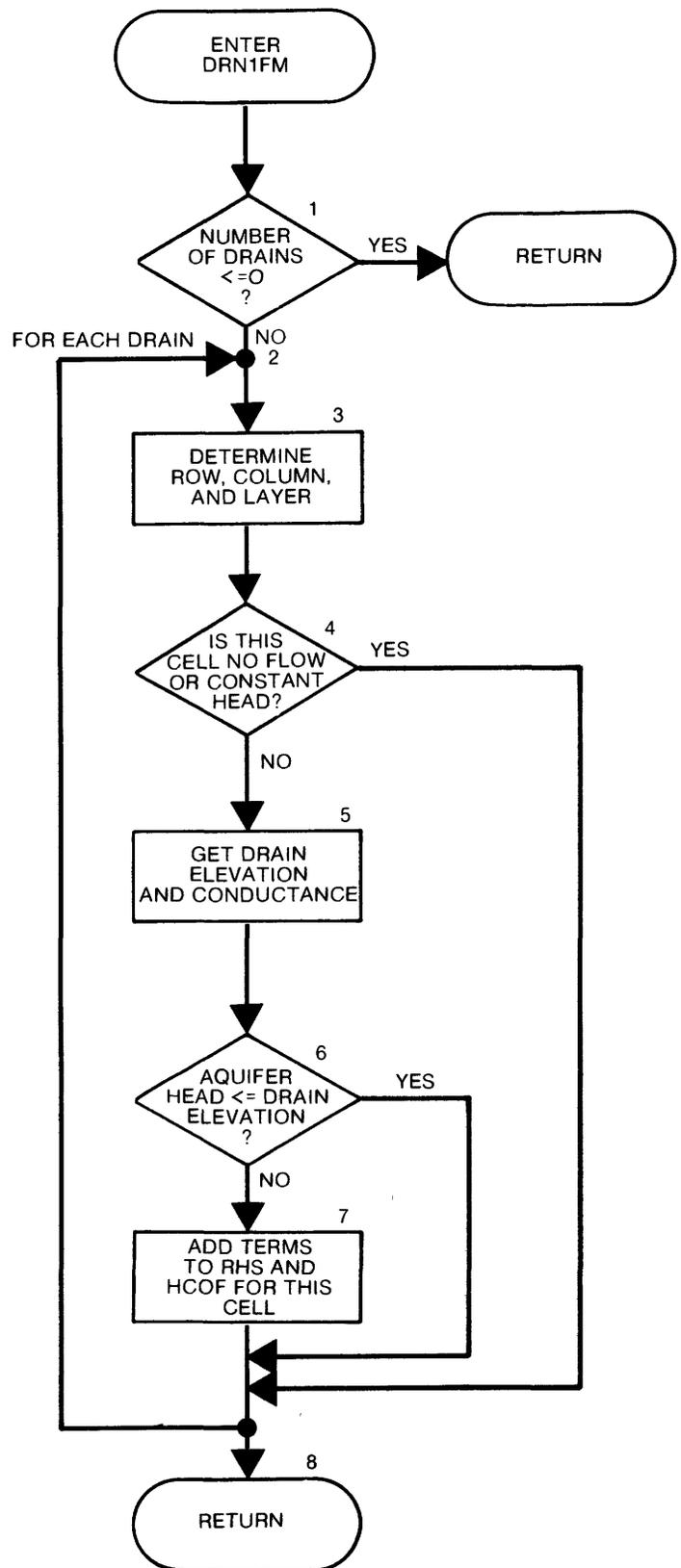
This module adds terms representing drain leakage to the accumulators HCOF and RHS.

1. If NDRAIN is less than or equal to zero in the current stress period, there are no drains. RETURN.
2. For each drain in the drain list, DO STEPS 3-7.
3. Determine the column (IC), row (IR), and layer (IL).
4. If the cell is external ( $IBOUND(IC, IR, IL) \leq 0$ ), bypass processing on this drain and go on to the next drain.
5. If the cell is internal, get the drain data (elevation and conductance).
6. If the head in the aquifer (HHNEW) is greater than the elevation of the drain, there is no drain leakage. RETURN.
7. If the head in the aquifer (HHNEW) is greater than the elevation of the drain (EL), add the term  $-C*EL$  (C is the drain conductance) to the accumulator RHS and the term  $-C$  to the accumulator HCOF.
8. RETURN.

Flow Chart for Module DRN1FM

RHS is an accumulator in which the right hand side of the equation is formulated.

HCOF is an accumulator in which the coefficient of head in the cell is formulated.



```

      SUBROUTINE DRN1FM(NDRAIN,MXDRN,DRAI,HNEW,HCOF,RHS,IBOUND,
1          NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1638 25APR1983 DRN1FM
C
C *****
C ADD DRAIN FLOW TO SOURCE TERM
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION DRAI(5,MXDRN),HNEW(NCOL,NROW,NLAY),
1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
1          HCOF(NCOL,NROW,NLAY)
C -----
C
C1-----IF NDRAIN<=0 THERE ARE NO DRAINS. RETURN
          IF(NDRAIN.LE.0) RETURN
C
C2-----PROCESS EACH CELL IN THE DRAIN LIST
          DO 100 L=1,NDRAIN
C
C3-----GET COLUMN, ROW AND LAYER OF CELL CONTAINING DRAIN.
          IL=DRAI(1,L)
          IR=DRAI(2,L)
          IC=DRAI(3,L)
C
C4-----IF THE CELL IS EXTERNAL SKIP IT.
          IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C5-----IF THE CELL IS INTERNAL GET THE DRAIN DATA.
          EL=DRAI(4,L)
          HHNEW=HNEW(IC,IR,IL)
C
C6-----IF HEAD IS LOWER THAN DRAIN THEN SKIP THIS CELL.
          IF(HHNEW.LE.EL) GO TO 100
C
C7-----HEAD IS HIGHER THAN DRAIN. ADD TERMS TO RHS AND HCOF.
          C=DRAI(5,L)
          HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C
          RHS(IC,IR,IL)=RHS(IC,IR,IL)-C*EL
          100 CONTINUE
C
C8-----RETURN
          RETURN
          END

```

List of Variables for Module DRN1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
C	Module	Conductance into the drain.
DRAI	Package	DIMENSION (5,MXDRN), For each drain: layer, row, column, head in the drain and conductance into the drain.
EL	Module	Elevation of the drain (head in the drain).
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HHNEW	Module	Head in the cell containing the drain.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
MXDRN	Package	Maximum number of drains active at any one time.
NCOL	Global	Number of columns in the grid.
NDRAIN	Package	Number of drains active during the current stress period.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

### Narrative for Module DRN1BD

This module calculates rates and volumes transferred between the aquifer and drains.

1. Initialize the cell-by-cell flow-term flag (IBD) and the rate accumulator (RATOUT).

2. If there are no drains ( $NDRAIN \leq 0$ ), skip down to step 12 and put zeros into the budget terms for drains.

3. Test to see if cell-by-cell flow terms are to be saved on disk. They will not be saved if either of the following conditions hold: (1) this is not the proper time step ( $ICBCFL = 0$ ) or (2) cell-by-cell flow terms are not needed for drains during this simulation ( $IDRNCB \leq 0$ ). If cell-by-cell flow terms will be saved for drains, set the cell-by-cell flow-term flag (IBD) and clear the buffer in which they will be accumulated (BUFF).

4. For each drain, do steps 3-11 accumulating flows into drains.

5. Determine the row, column, and layer of the cell containing the drain.

6. If the cell is external ( $IBOUND(I,J,K) \leq 0$ ), bypass further processing of this drain.

7. Get the drain parameters from the drain list.

8. If the head in the cell is less than the elevation of the drain, bypass further processing of this drain.

9. If the head in the cell is greater than the elevation of the drain, set "Q" equal to the conductance of the drain (C) times the drain elevation (EL) minus the head in the cell (HHNEW) ( $Q = C*(EL - HHNEW)$ ). Add Q to the accumulator RATOUT to get the total flow from the aquifer into drains.

10. If the cell-by-cell flow terms are to be printed ( $IDRNCB < 0$  and  $ICBCFL \neq 0$ ), print Q.

11. If the cell-by-cell flow terms for drains are to be saved, add Q to the buffer (BUFF).

12. See if the cell-by-cell flow terms are to be saved ( $IBD = 1$ ). If they are, call module UBUDSV to record the buffer (BUFF) onto disk.

13. Move RATOUT into the VBVL array for printing by BAS10T. Add RATOUT multiplied by the time-step length to the volume accumulator in VBVL for printing by BAS10T. Move the drain budget-term labels to VBNM for print by BAS10T.

14. Increment the budget-term counter (MSUM). See the section in the Basic Package for a detailed explanation of VBVL, VBNM, and MSUM.

15. RETURN.

## Flow Chart for Module DRN1BD

IBD is a flag which, if set, causes cell-by-cell flow terms for drains to be recorded.

EXTERNAL: a cell is said to be external if it is either no flow or constant head (i.e., an equation is not formulated for the cell).

BUFFER is an array in which values are stored as they are being gathered for printing or recording.

RATOUT is an accumulator to which all flows out of the aquifer are added.

Q is the discharge to a drain.

EL is the elevation of the drain.

IDRNCB is a flag and a unit number.

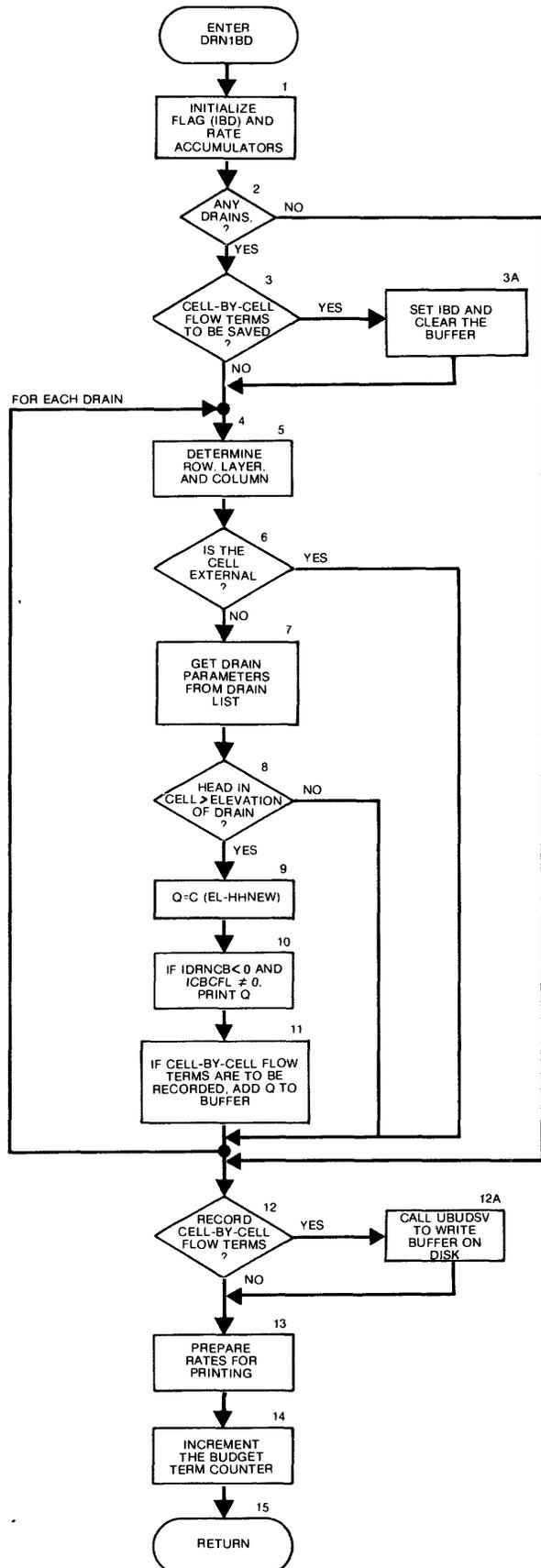
If  $IDRNCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If  $IDRNCB = 0$ , cell-by-cell flow terms will not be printed or recorded.

If  $IDRNCB < 0$ , drain leakage for each drain will be printed whenever ICBCFL is set.

ICBCFL is a flag.

If  $ICBCFL \neq 0$ , cell-by-cell flow terms will be either printed or recorded for the current time step.



```

SUBROUTINE DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,DRAI,DELT,HNEW,
1          NCOL,NROW,NLAY,IBOUND,KSTP,KPER,IDRNCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1301 28DEC1983 DRN1BD
C
C *****
C CALCULATE VOLUMETRIC BUDGET FOR DRAINS
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C
C DIMENSION VBNM(4,MSUM),VBVL(4,MSUM),DRAI(5,MXDRN),
1          HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
2          BUFF(NCOL,NROW,NLAY)
C DIMENSION TEXT(4)
C
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' ',' ','DR','AINS'/
C -----
C1-----INITIALIZE CELL-BY-CELL FLOW TERM FLAG (IBD) AND
C1-----ACCUMULATORS (RATIN AND RATOUT).
C          RATOUT=0.
C          IBD=0
C
C2-----IF THERE ARE NO DRAINS THEN DO NOT ACCUMULATE DRAIN FLOW
C          IF(NDRAIN.LE.0) GO TO 200
C
C3-----TEST TO SEE IF CELL-BY-CELL FLOW TERMS ARE NEEDED.
C          IF(ICBCFL.EQ.0 .OR. IDRNCB.LE.0) GO TO 60
C
C3B-----CELL-BY-CELL FLOW TERMS ARE NEEDED SET IBD AND CLEAR BUFFER.
C          IBD=1
C          DO 50 IL=1,NLAY
C          DO 50 IR=1,NROW
C          DO 50 IC=1,NCOL
C          BUFF(IC,IR,IL)=0.
C          50 CONTINUE
C
C4-----FOR EACH DRAIN ACCUMULATE DRAIN FLOW
C          60 DO 100 L=1,NDRAIN
C
C5-----GET LAYER, ROW & COLUMN OF CELL CONTAINING REACH.
C          IL=DRAI(1,L)
C          IR=DRAI(2,L)
C          IC=DRAI(3,L)
C
C6-----IF CELL IS EXTERNAL IGNORE IT.
C          IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C7-----GET DRAIN PARAMETERS FROM DRAIN LIST.
C          EL=DRAI(4,L)
C          C=DRAI(5,L)

```

```

      HHNEW=HNEW(IC,IR,IL)
C
C8-----IF HEAD LOWER THAN DRAIN THEN FORGET THIS CELL.
      IF(HHNEW.LE.EL) GO TO 100
C
C9-----HEAD HIGHER THAN DRAIN. CALCULATE Q=C*(EL-HHNEW) ADD Q TO RATOUT
      Q=C*(EL-HHNEW)
      RATOUT=RATOUT+Q
C
C10-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IDRNCB<0).
      IF(IDRNCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
1      KPER,KSTP,L,IL,IR,IC,Q
900 FORMAT(1H0,4A4,' PERIOD',I3,' STEP',I3,' DRAIN',I4,
1      ' LAYER',I3,' ROW',I4,' COL',I4,' RATE',G15.7)
C
C11-----IF C-B-C FLOW TERMS ARE TO BE SAVED THEN ADD Q TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+Q
100 CONTINUE
C
C12-----IF C-B-C FLOW TERMS WILL BE SAVED CALL UBUDSV TO RECORD THEM.
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IDRNCB,BUFF,NCOL,NROW,
1      NLAY,IOUT)
C
C13-----MOVE RATES,VOLUMES & LABELS INTO ARRAYS FOR PRINTING.
200 VBVL(3,MSUM)=0.
      VBVL(4,MSUM)=RATOUT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C14-----INCREMENT BUDGET TERM COUNTER
      MSUM=MSUM+1
C
C15-----RETURN
      RETURN
      END

```

List of Variables for Module DRN1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
C	Module	Conductance into drains.
DELTA	Global	Length of the current time step.
DRAI	Package	DIMENSION (5,MXDRN), For each drain: layer, row, column, head in the drain and conductance into the drain.
EL	Module	Elevation of the drain (head in the drain).
HHNEW	Module	Head in the cell containing the drain.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBD	Package	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
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 for the current time step.
IDRNCB	Package	Flag. > 0 and if ICBCFL ≠ 0, cell-by-cell flow terms for the DRN1 Package will be recorded on UNIT = IDRNCB.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
KPER	Global	Stress period counter.

List of Variables for Module DRN1BD (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for drains.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXDRN	Package	Maximum number of drains active at any one time.
NCOL	Global	Number of columns in the grid.
NDRAIN	Package	Number of drains active during the current stress period.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
Q	Module	Flow from a drain into a cell. (Reverse the sign to get the flow into the drain.)
RATOUT	Module	Accumulator for the total flow out of the flow field into the drains.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N) Rate for the current time step into the flow field. (2,N) Rate for the 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.

## CHAPTER 10

### EVAPOTRANSPIRATION PACKAGE

#### Conceptualization and Implementation

Evapotranspiration (ET) is the mechanism whereby water is converted from the liquid phase to the vapor phase. This package simulates the effect of ET where the source of water is the saturated porous medium; therefore, it deals primarily with water removed by the roots of plants.

The ET rate determined by the ET Package depends on the position of the aquifer head relative to two given ET reference elevations--ET surface and ET extinction elevation (fig. 42). For a given model node, the ET rate decreases to zero as the aquifer head declines to the extinction elevation and is set to zero when the aquifer head drops below this elevation. The ET rate increases to a user-controlled maximum limit as the aquifer head rises above the extinction elevation to the given ET surface elevation. The ET rate is assumed to be proportional to the saturated thickness above the given ET extinction elevation. The ET rate is expressed in terms of flow into the aquifer as

$$Q = 0 \text{ when } h < \text{EXEL} \quad (70)$$

$$Q = \text{EVTR} (h - \text{EXEL})/\text{EXDP} \text{ when } \text{SURF} \geq h \geq \text{EXEL} \quad (71)$$

$$Q = \text{EVTR} \text{ when } h > \text{SURF} \quad (72)$$

where

Q is the ET rate ( $L^3t^{-1}$ );

h is the head in the aquifer (L);

EXEL is the extinction elevation (L);

SURF is the ET surface elevation (L);

EXDP is the extinction depth ( $\text{SURF} - \text{EXEL}$ ) (L); and

EVTR is the maximum ET rate ( $L^3t^{-1}$ ).

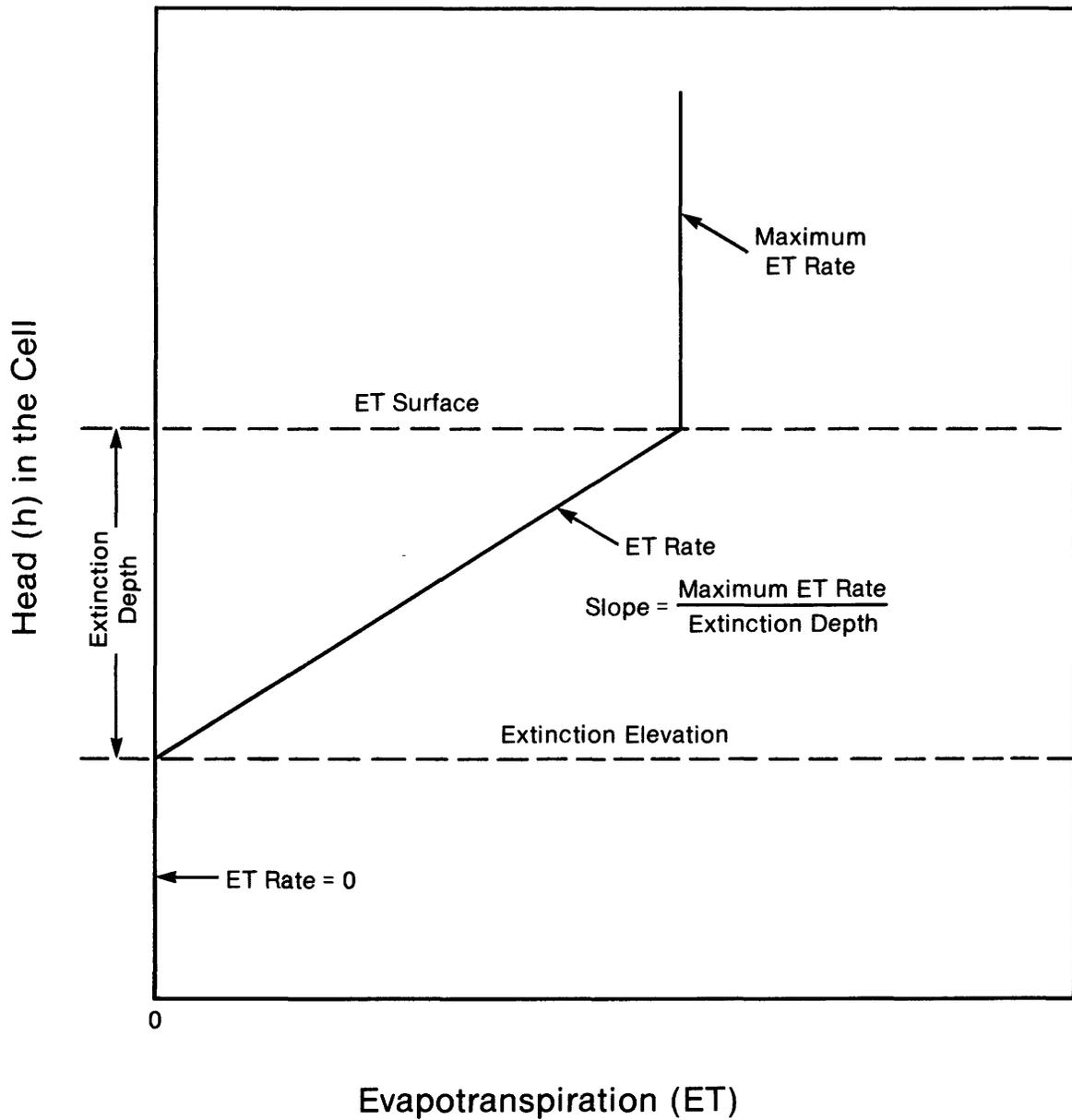


Figure 42.—Evapotranspiration as a function of head in the aquifer.

Replacing the term EXEL by the expression (SURF - EXDP) in equations 70, 71, and 72 yields

$$Q = 0 \text{ when } h < \text{SURF} - \text{EXDP} \quad (73)$$

$$Q = \text{EVTR} (h - (\text{SURF} - \text{EXDP}))/\text{EXDP} \text{ when } \text{SURF} \geq h \geq \text{SURF} - \text{EXDP} \quad (74)$$

$$Q = \text{EVTR} \text{ when } h > \text{SURF}. \quad (75)$$

To simulate the effect of ET on an aquifer, one of these expressions is added to the finite-difference equation for each cell. The finite-difference equation was written in the form

$$\begin{aligned} & CV_{i,j,k-1/2} h_{i,j,k-1}^m + CC_{i-1/2,j,k} h_{i-1,j,k}^m + CR_{i,j-1/2,k} h_{i,j-1,k}^m \\ & + (- CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\ & - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + \text{HCOF}_{i,j,k}) h_{i,j,k}^m + CR_{i,j+1/2,k} h_{i,j+1,k}^m \\ & + CC_{i+1/2,j,k} h_{i+1,j,k}^m + CV_{i,j,k+1/2} h_{i,j,k+1}^m = \text{RHS}_{i,j,k} \end{aligned} \quad (76)$$

where

$\text{RHS}_{i,j,k}$  is composed of all terms independent of head at the end of the iteration; and

$\text{HCOF}_{i,j,k}$  is composed of all coefficients of  $h_{i,j,k}$  other than conductances from adjacent cells.

As explained in chapter 2, an external source of water represented by an equation of the form

$$a_{i,j,k,n} = P_{i,j,k,n} h_{i,j,k} + Q_{i,j,k,n} \quad (77)$$

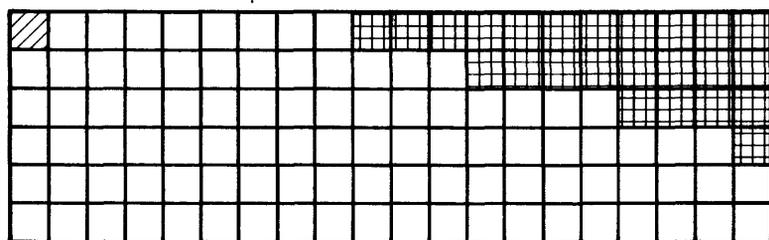
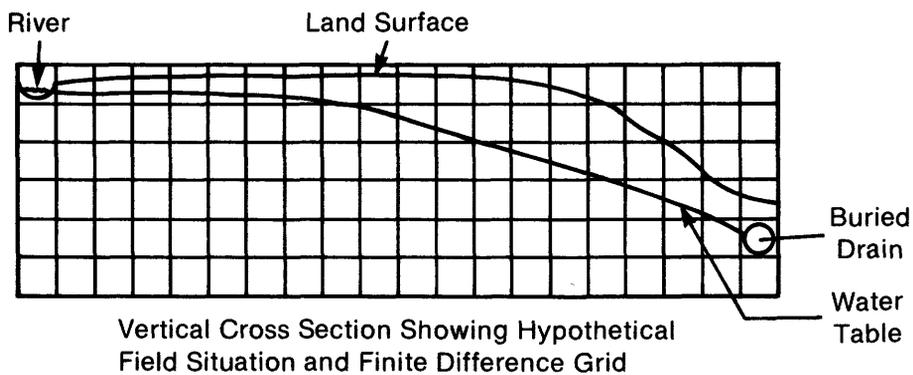
can be incorporated into equation 76 by adding  $P_{i,j,k,n}$  to  $\text{HCOF}_{i,j,k}$  and subtracting  $Q_{i,j,k,n}$  from  $\text{RHS}_{i,j,k}$ . The value of  $P_{i,j,k,n}$  and  $Q_{i,j,k,n}$

depends on which of the three equations--73, 74, or 75 is selected. The values are shown in the following table.

	Equation		
	73	74	75
$P_{i,j,k,n}$	0	$-EVTR/EXDP$	0
$Q_{i,j,k,n}$	0	$EVTR*(SURF-EXDP)/EXDP$	$-EVTR$

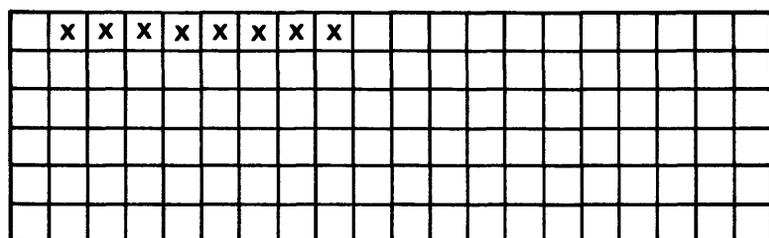
The data needed to calculate the terms in this table are the three variables--maximum ET rate, ET surface elevation, and extinction depth. The three variables are stored in the three arrays--EVTR, SURF, and EXDP, each array having one value for each horizontal-cell location. The user must specify values for each variable at each horizontal-cell location.

There are two options (fig. 43) for indicating from which layer ET is abstracted at a given horizontal-cell location. Under option 1, ET is taken from grid layer 1. Under option 2, ET is taken from the layer specified by the user in a special array called IEVT.



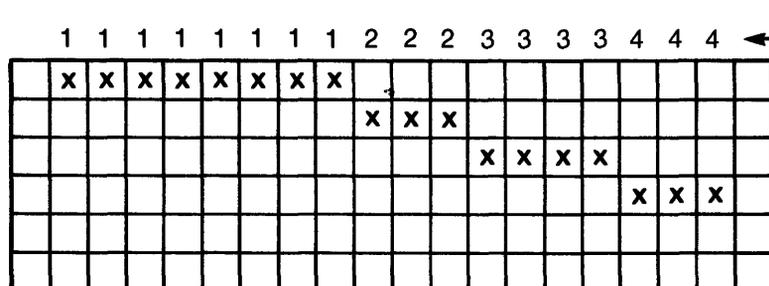
Status of Cells at End of Simulated Period

- Variable Head Cell
- Constant Head Cell
- Inactive Cell



Cells from Which ET Is Abstracted Under Option 1

- Cell from Which ET Is Abstracted



Cells from Which ET Is Abstracted Under Option 2

Layer Indicators Specified in the IEVT Array

- Cell from Which ET Is Abstracted

Figure 43.—Hypothetical problem showing from which cells ET will be abstracted under the two options available in the ET Package.

## Evapotranspiration Package Input

Input to the Evapotranspiration (EVT) Package is read from the unit specified in IUNIT (5).

FOR EACH SIMULATION

### EVT1AL

- |    |         |        |        |
|----|---------|--------|--------|
| 1. | Data:   | NEVTOP | IEVTCB |
|    | Format: | I10    | I10    |

FOR EACH STRESS PERIOD

### EVT1RP

- |    |         |        |        |        |        |
|----|---------|--------|--------|--------|--------|
| 2. | Data:   | INSURF | INEVTR | INEXDP | INIEVT |
|    | Format: | I10    | I10    | I10    | I10    |
| 3. | Data:   | SURF   |        |        |        |
|    | Module: | U2DREL |        |        |        |
| 4. | Data:   | EVTR   |        |        |        |
|    | Module: | U2DREL |        |        |        |
| 5. | Data:   | EXDP   |        |        |        |
|    | Module: | U2DREL |        |        |        |

IF THE ET OPTION IS EQUAL TO TWO

- |    |         |        |
|----|---------|--------|
| 6. | Data:   | IEVT   |
|    | Module: | U2DINT |

### Explanation of Fields Used in Input Instructions

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

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

IEVTCB--is a flag and a unit number.

If  $IEVTCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If  $IEVTCB \leq 0$ , cell-by-cell flow terms will not be printed or recorded.

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

If  $INSURF \geq 0$ , an array containing the ET surface elevation will be read.

If  $INSURF < 0$ , the ET surface from the preceding stress period will be reused.

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

If  $INEVTR \geq 0$ , an array containing the maximum ET rate will be read.

If  $INEVTR < 0$ , the maximum ET rate from the preceding stress period will be reused.

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

If  $INEXDP \geq 0$ , an array containing the extinction depth (EXDP) will be read.

If  $INEXDP < 0$ , the extinction depth from the preceding stress period will be reused.

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

If  $INIEVT \geq 0$ , an array containing the layer indicators (IEVT) will be read.

If  $INIEVT < 0$ , layer indicators used during the preceding stress period will be reused.

SURF--is the elevation of the ET surface.

EVTR--is the maximum ET rate.

EXDP--is the ET extinction depth.

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

SAMPLE INPUT TO THE EVAPOTRANSPIRATION PACKAGE USING ET OPTION 1

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{NEVTOP, IEVTCB}	1 0
2	Stress period 1--{INSURF, INEVTR, INEXDP, INIEVT}	1 1 1
3	Control record for ET surface array	27 1. (10F5.0)
	ET surface	710 715 720 725 730
		715 720 725 730 735
		720 725 730 735 740
		725 730 735 740 745
		730 735 740 745 750
		735 740 745 750 755
4	Control record for maximum ET rate	0 9.65E-7
5	Control record for extinction depth array	0 10.
2	Stress period 2--{INSURF, INEVTR, INEXDP, INIEVT}	-1 -1
4	Control record for maximum ET rate	0 8.23E-7
2	Stress period 3--{INSURF, INEVTR, INEXDP, INIEVT}	-1 -1
4	Control record for maximum ET rate	27 9.65E-7 (10F4.0)
	max ET rate	1.2 1.2 1.2 1.2 1.3
		1.2 1.2 1.2 1.3 1.4
		1.2 1.2 1.3 1.4 1.4
		1.0 1.0 1.0 1.1 1.1
		1.2 1.3 1.3 1.4 1.4
		1.3 1.3 1.4 1.4 1.4

3 2 3

SAMPLE INPUT TO THE EVAPOTRANSPIRATION PACKAGE USING ET OPTION 2

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{NEVTOP, IEVTCB}	2 45
2	Stress period 1--{INSURF, INEVTR, INEXDP, INIEVT}	1 1 1
3	Control record for ET surface array	27 1. (10F5.0)
	ET surface	710 715 720 725 730
		715 720 725 730 735
		720 725 730 735 740
		725 730 735 740 745
		730 735 740 745 750
		735 740 745 750 755
4	Control record for maximum ET rate	0 9.65E-7
5	Control record for extinction depth array	0 10.
6	Control record for layer indicator array	12 1 (2012)
	Layer numbers	1 2 2 2 3
		1 2 2 2 2
		1 1 2 2 2
		1 1 1 1 2
		1 1 1 1 1
		1 1 1 1 1
		1 1 1 1 1
2	Stress period 2--{INSURF, INEVTR, INEXDP, INIEVT}	-1 -1
4	Control record for maximum ET rate	0 8.23E-7

FIELDS IN ARRAY CONTROL RECORDS ARE---{ LOCAT, CONST, FMTIN, IPRN}

## Module Documentation for the Evapotranspiration Package

The Evapotranspiration Package (EVT1) consists of four modules, all of which are called by the MAIN program. The modules are:

- EVT1AL      Allocates space for arrays to contain maximum ET rate (EVTR), surface elevation (SURF), extinction depth (EXDP), and, if option 2 is specified, the layer indicator (IEVT).
- EVT1RP      Reads arrays containing the maximum ET rate (in terms of a volume per unit area), surface elevation, extinction depth, and, if option 2 is specified, the layer indicator. Maximum ET rates are multiplied by cell area to get the maximum ET for each node as a volumetric rate.
- EVT1FM      Determines, for each horizontal location, which cell is at the surface. Determines if there is ET from that cell. If there is ET, add the appropriate terms to HCOF(I,J,K) and RHS(I,J,K).
- EVT1BD      Calculates the rates and accumulated volume of ET out of the flow system.

### Narrative for Module EVT1AL

This module allocates space in the X array to store data relating to evapotranspiration.

1. Print a message identifying the package.
2. Read and print the option indicator (NEVTOP) and the unit number for cell-by-cell flow terms (IEVTCB).
3. See if the ET option (NEVTOP) is legal. If NEVTOP is illegal (not 1 or 2), print a message saying the option is illegal. Do not allocate storage. STOP.
4. If NEVTOP is legal, print NEVTOP.
5. If the cell-by-cell flow terms are to be recorded, print the unit number (IEVTCB) where they will be recorded.
6. Allocate space for the maximum ET-rate array (EVTR), the extinction-depth array (EXDP), and the ET-surface array (SURF).
7. If the ET option (NEVTOP) is equal to two, allocate space for a layer-indicator array (IEVT).
8. Calculate and print the number of elements in the X array used by the ET package.
9. RETURN.

Flow Chart for Module EVT1AL

NEVTOP is the ET option.

If NEVTOP = 1, ET is from the top layer.

If NEVTOP = 2, ET is from the layer specified by the user in the indicator array (IEVT).

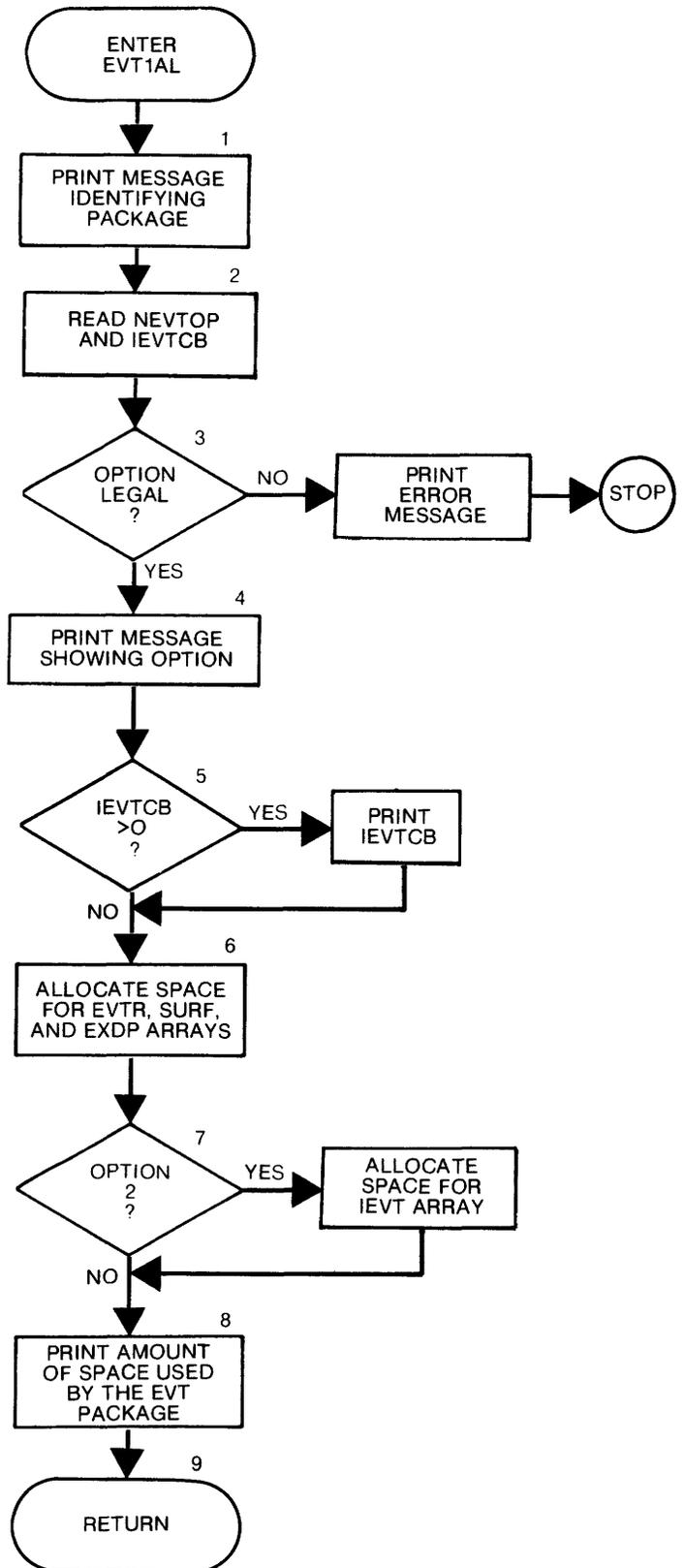
IEVTCB is the unit number on which cell-by-cell flow terms for ET will be written.

EVTR is an array which contains the maximum ET rate for each horizontal cell location.

SURF is an array which contains the elevation of the ET surface.

EXDP is an array which contains the extinction depth for ET.

IEVT is an array which contains the layer number from which ET is taken for each horizontal location. It is used only if option 2 has been specified.



```

SUBROUTINE EVTIAL (ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,LCSURF,
1 NCOL,NROW,NEVTOP,IN,IOUT,IEVTCB)
C
C-----VERSION 0943 08DEC1983 EVTIAL
C *****
C ALLOCATE ARRAY STORAGE FOR EVAPOTRANSPIRATION
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----IDENTIFY PACKAGE.
WRITE(IOUT,1)IN
1 FORMAT(1H0,'EVT1 -- EVAPOTRANSPIRATION PACKAGE, VERSION 1,',
1 ' 12/08/83',' INPUT READ FROM UNIT',I3)
C
C2-----READ NEVTOP AND IEVTCB.
READ(IN,3)NEVTOP,IEVTCB
3 FORMAT(2I10)
C
C3-----CHECK TO SEE THAT ET OPTION IS LEGAL.
IF (NEVTOP.GE.1.AND.NEVTOP.LE.2)GO TO 200
C
C3A-----IF ILLEGAL PRINT A MESSAGE & ABORT SIMULATION.
WRITE(IOUT,8)
8 FORMAT(1X,'ILLEGAL ET OPTION CODE. SIMULATION ABORTING')
STOP
C
C4-----IF THE OPTION IS LEGAL THEN PRINT THE OPTION CODE.
200 IF (NEVTOP.EQ.1) WRITE(IOUT,201)
201 FORMAT(1X,'OPTION 1 -- EVAPOTRANSPIRATION FROM TOP LAYER')
IF (NEVTOP.EQ.2) WRITE(IOUT,202)
202 FORMAT(1X,'OPTION 2 -- EVAPOTRANSPIRATION FROM ONE SPECIFIED',
1 ' NODE IN EACH VERTICAL COLUMN')
IRK=ISUM
C
C5-----IF CELL-BY-CELL TERMS TO BE SAVED THEN PRINT UNIT NUMBER.
IF (IEVTCB.GT.0) WRITE(IOUT,203) IEVTCB
203 FORMAT(1X,'CELL-BY-CELL FLOW TERMS WILL BE SAVED ON UNIT',I3)
C
C6-----ALLOCATE SPACE FOR THE ARRAYS EVTR, EXDP AND SURF.
LCEVTR=ISUM
ISUM=ISUM+NCOL*NROW
LCEXDP=ISUM
ISUM=ISUM+NCOL*NROW
LCSURF=ISUM
ISUM=ISUM+NCOL*NROW
C
C7-----IF OPTION 2 THEN ALLOCATE SPACE FOR THE INDICATOR ARRAY(IEVT)
IF (NEVTOP.NE.2)GO TO 300
LCIEVT=ISUM
ISUM=ISUM+NCOL*NROW
C
C8-----CALCULATE & PRINT AMOUNT OF SPACE USED BY ET PACKAGE.
300 IRK=ISUM-IRK
WRITE(IOUT,4)IRK
4 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED FOR EVAPOTRANSPIRATION')
ISUM1=ISUM-1
WRITE(IOUT,5)ISUM1,LENX
5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
IF (ISUM1.GT.LENX)WRITE(IOUT,6)
6 FORMAT(1X,' ***X ARRAY MUST BE MADE LARGER***')
C
C9-----RETURN.
RETURN
END

```

List of Variables for Module EVT1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IEVTCB	Package	Flag. If IEVTCB > 0 and ICBCFL ≠ 0, cell-by-cell flow terms for the EVT1 Package will be recorded on UNIT = IEVTCB.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IRK	Module	Before this module allocates space, IRK is set equal to ISUM. After allocation, IRK is subtracted from ISUM to get the amount of space in the X array allocated by this module.
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.
LCEVTR	Package	Location in the X array of the first element of array EVTR.
LCXDP	Package	Location in the X array of the first element of array EXDP.
LCIEVT	Package	Location in the X array of the first element of array IEVT.
LCSURF	Package	Location in the X array of the first element of array SURF.
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.
NEVTOP	Package	ET option: = 1, ET is from the top layer. = 2, ET at each horizontal-cell location is from the layer-indicator array (IEVT).
NROW	Global	Number of rows in the grid.

## Narrative for Module EVT1RP

This module reads data used to calculate the terms which represent evapotranspiration.

1. Read the values INSURF, INEXDP, INEVTR, and INIEVT which indicate whether the data contained in arrays SURF, EXDP, EVTR, and IEVT, respectively, used during the last stress period, are to be used for the current stress period.

2. Test INSURF to see where the ET-surface array (SURF) is coming from. If INSURF is less than zero, the ET-surface elevation used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 4.

3. INSURF is greater than or equal to zero. CALL U2DREL to read SURF.

4. Test INEVTR to see where the maximum ET rate (EVTR) is coming from. If INEVTR is less than zero, the maximum ET rate used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 7.

5. INEVTR is greater than or equal to zero. CALL U2DREL to read the maximum ET rate (EVTR).

6. Multiply the maximum ET rate by the area to get a volumetric rate.

7. Test INEXDP to see where the extinction rate is coming from. If INEXDP is less than zero, the extinction depth used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 9.

8. If INEXDP is greater than or equal to zero, CALL U2DREL to read the extinction depth.

9. If the ET option (NEVTOP) is equal to two, a layer-indicator array is needed.

10. Test INIEVT to see where the layer indicator is coming from. If INIEVT is less than zero, the indicator array used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 12.

11. If INIEVT is greater than or equal to zero, CALL U2DINT to read the IEVT array.

12. RETURN.

## Flow Chart for Module EVT1RP

INEVTR is a flag which, when set, indicates that the maximum ET rate EVTR should be read for the current stress period. If it is clear (less than zero), maximum ET rates from the last stress period should be reused.

INIEVT, INSURF, and INEXDP are flags similar to INEVTR used for the layer indicator array (IEVT), the ET surface array (SURF), and the extinction depth array (EXDP), respectively.

EVTR is an array containing the maximum ET rate for every horizontal cell location.

SURF is an array containing the ET surface elevation for each horizontal cell location.

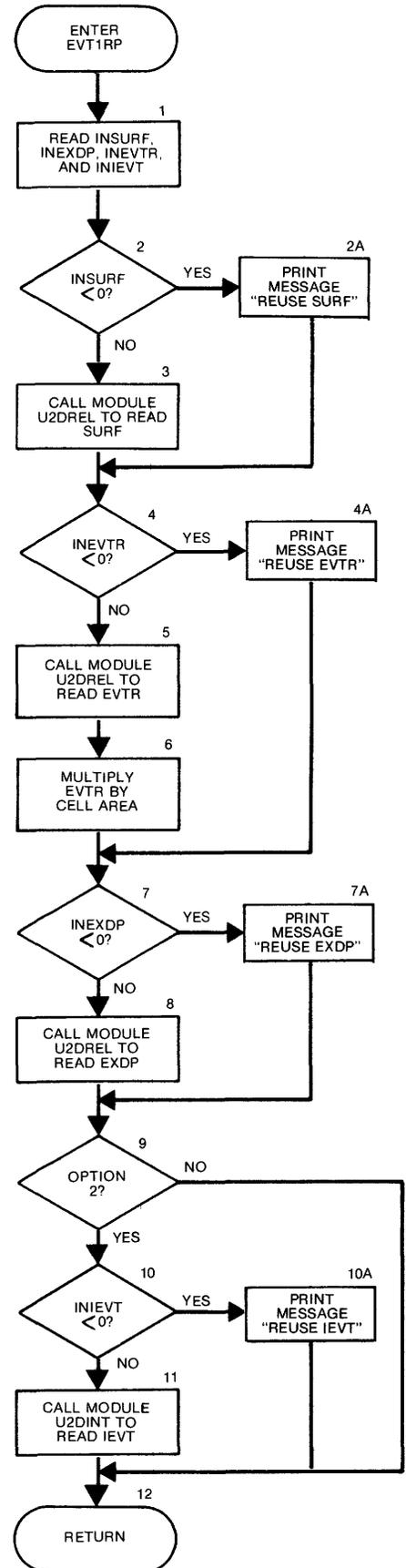
EXDP is an array containing the extinction depth for each horizontal cell location.

IEVT is an array containing a layer indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location from which ET is taken. It is used only if the ET option (NEVTOP) is equal to two.

NEVTOP is the ET option.

If NEVTOP = 1, ET is from the top layer.

If NEVTOP = 2, ET is from the layer specified by the user in the indicator array (IEVT).



```

SUBROUTINE EVTIRP(NEVTOP,IEVT,EVTR,EXDP,SURF,DELR,DELC,
1              NCOL,NROW,NLAY,IN,IOUT)
C
C-----VERSION 1631 08FEB1983 EVTIRP
C *****
C READ EVAPOTRANSPIRATION DATA
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
1 SURF(NCOL,NROW),ANAME(6,4),DELR(NCOL),DELC(NROW)
C
C DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
1 ANAME(6,1) / ' ' ' ' ' ' ' ' ET ' ' LAY ' ' ER I ' ' NDEX ' /
C DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
1 ANAME(6,2) / ' ' ' ' ' ' ' ' ET ' ' SUR ' ' FACE ' /
C DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
1 ANAME(6,3) / ' EVA ' ' POTR ' ' ANSP ' ' IRAT ' ' ION ' ' RATE ' /
C DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),
1 ANAME(6,4) / ' ' ' ' ' ' ' ' EXT I ' ' NCT I ' ' ON D ' ' EPTH ' /
C -----
C
C1-----READ FLAGS SHOWING WHETHER DATA IS TO BE REUSED.
READ(IN,6)INSURF,INEVTR,INEXDP,INIEVT
6 FORMAT(4I10)
C
C2-----TEST INSURF TO SEE WHERE SURFACE ELEVATION COMES FROM.
IF(INSURF.GE.0)GO TO 32
C
C2A-----IF INSURF<0 THEN REUSE SURFACE ARRAY FROM LAST STRESS PERIOD
WRITE(IOUT,3)
3 FORMAT(1H0,'REUSING SURF FROM LAST STRESS PERIOD')
GO TO 35
C
C3-----IF INSURF=>0 THEN CALL MODULE U2DREL TO READ SURFACE.
32 CALL U2DREL(SURF,ANAME(1,2),NROW,NCOL,0,IN,IOUT)
C
C4-----TEST INEVTR TO SEE WHERE MAX ET RATE COMES FROM.
35 IF(INEVTR.GE.0)GO TO 37
C
C4A-----IF INEVTR<0 THEN REUSE MAX ET RATE.
WRITE(IOUT,4)
4 FORMAT(1H0,'REUSING EVTR FROM LAST STRESS PERIOD')
GO TO 45
C
C5-----IF INEVTR=>0 CALL MODULE U2DREL TO READ MAX ET RATE.
37 CALL U2DREL(EVTR,ANAME(1,3),NROW,NCOL,0,IN,IOUT)
C
C6-----MULTIPLY MAX ET RATE BY CELL AREA TO GET VOLUMETRIC RATE
DO 40 IR=1,NROW
DO 40 IC=1,NCOL
EVTR(IC,IR)=EVTR(IC,IR)*DELR(IC)*DELC(IR)
40 CONTINUE
C
C7-----TEST INEXDP TO SEE WHERE EXTINCTION DEPTH COMES FROM
45 IF(INEXDP.GE.0)GO TO 47
C
C7A-----IF INEXDP<0 REUSE EXTINCTION DEPTH FROM LAST STRESS PERIOD
WRITE(IOUT,5)
5 FORMAT(1H0,'REUSING EXDP FROM LAST STRESS PERIOD')
GO TO 48
C
C8-----IF INEXDP=>0 CALL MODULE U2DREL TO READ EXTINCTION DEPTH
47 CALL U2DREL(EXDP,ANAME(1,4),NROW,NCOL,0,IN,IOUT)
C
C9-----IF OPTION(NEVTOP) IS 2 THEN WE NEED AN INDICATOR ARRAY.
48 IF(NEVTOP.NE.2)GO TO 50
C
C10-----IF INIEVT<0 THEN REUSE LAYER INDICATOR ARRAY.
IF(INIEVT.GE.0)GO TO 49
WRITE(IOUT,2)
2 FORMAT(1H0,'REUSING IEVT FROM LAST STRESS PERIOD')
GO TO 50
C
C11-----IF INIEVT=>0 THEN CALL MODULE U2DINT TO READ INDICATOR ARRAY.
49 CALL U2DINT(IEVT,ANAME(1,1),NROW,NCOL,0,IN,IOUT)
C
C12-----RETURN
50 RETURN
END

```

List of Variables for Module EVT1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ANAME	Module	Label for printout of the input array.
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
EVTR	Package	DIMENSION (NCOL,NROW), Maximum ET rate.
EXDP	Package	DIMENSION (NCOL,NROW), Extinction depth.
IC	Module	Index for columns.
IEVT	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal cell location from which ET will be taken if the ET option (NEVTOP) is equal to two.
IN	Package	Primary unit number from which input for this package will be read.
INEVTR	Module	Flag. $\geq 0$ , EVTR array will be read. $< 0$ , EVTR array already in memory from the last stress period will be used.
INEXDP	Module	Flag. $\geq 0$ , EXDP array will be read. $< 0$ , EXDP array already in memory from the last stress period will be used.
INIEVT	Module	Flag. $\geq 0$ , IEVT array will be read. $< 0$ , IEVT array already in memory from the last stress period will be used.
INSURF	Module	Flag. $\geq 0$ , SURF array will be read. $< 0$ , SURF array already in memory from the last stress period will be used.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option. = 1, ET is from the top layer. = 2, ET at each horizontal-cell location is from the layer-indicator array (IEVT).
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
SURF	Package	DIMENSION (NCOL,NROW), Elevation of the ET surface.

### Narrative for Module EVT1FM

This module adds terms representing ET to the finite-difference equations.

1. For each horizontal-cell location, determine which layer ET comes from and add the appropriate terms to the equation for the cell. DO STEPS 1-7.

2. Set the layer index equal to one.

3. If option 2 was invoked, get the layer index from the indicator array (IEVT).

4. If the cell is external, move on to the next horizontal-cell location. SKIP STEPS 5-7.

5. If the head in the aquifer is greater than or equal to the ET-surface elevation, add EVTR to RHS and move on to the next horizontal-cell location. SKIP STEPS 6 AND 7.

6. If the head in the aquifer is less than the extinction elevation (ET surface minus extinction depth), no terms need to be added to the finite-difference equation. Move on to the next horizontal-cell location. SKIP STEP 7.

7. Add the term  $-EVTR/EXDP$  to HCOF and subtract the term  $-EVTR(EXDP - SURF)/EXDP$  from RHS.

8. RETURN.

## Flow Chart for Module EVT1FM

IEVT is an array containing a layer indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location from which ET is taken. It is used only if the ET option (NEVTOP) is equal to two.

SURF is an array containing the maximum ET rate for every horizontal cell location.

EVTR is an array containing the maximum ET rate for every horizontal cell location.

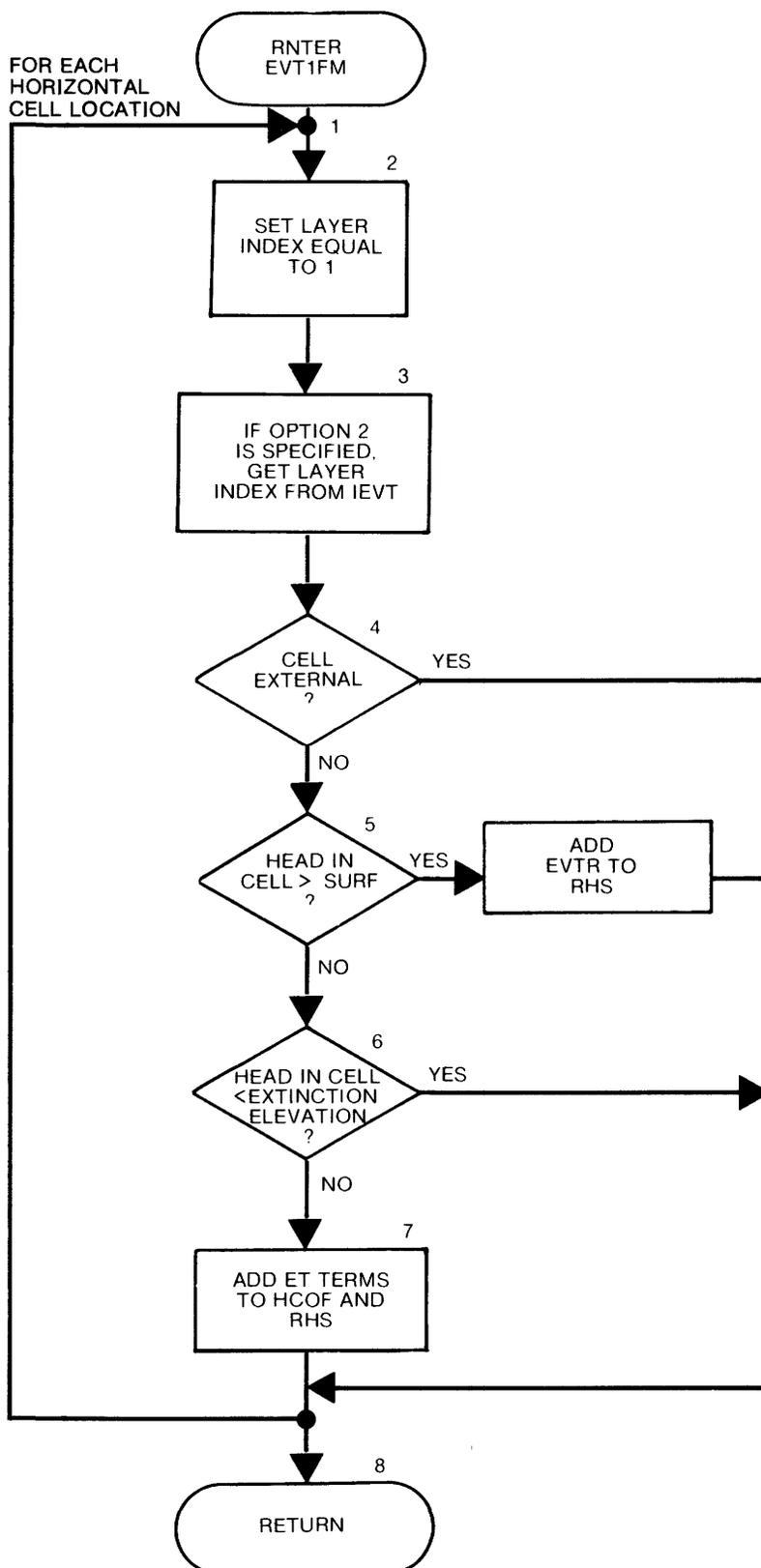
RHS is an accumulator in which the right hand side of the finite-difference equation is formulated.

HCOF is an accumulator in which a coefficient of head in the finite-difference equation is formulated.

NEVTOP is the ET option.

If NEVTOP = 1, ET is from the top layer.

If NEVTOP = 2, ET is from the layer specified by the user in the indicator array (IEVT).



```

      SUBROUTINE EVT1FM(NEVTOP,IEVT,EVTR,EXDP,SURF,RHS,HCOF,
1          IBOUND,HNEW,NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 0835 10FEB1983 EVT1FM
C *****
C      ADD EVAPOTRANSPIRATION TO RHS AND HCOF
C *****
C
C      SPECIFICATIONS:
C -----
C      DOUBLE PRECISION HNEW
C      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
1          SURF(NCOL,NROW),RHS(NCOL,NROW,NLAY),
2          HCOF(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
3          HNEW(NCOL,NROW,NLAY)
C -----
C
C1-----PROCESS EACH HORIZONTAL CELL LOCATION
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C2-----SET THE LAYER INDEX EQUAL TO 1
      IL=1
C
C3-----IF OPTION 2 IS SPECIFIED THEN GET LAYER INDEX FROM IEVT ARRAY
      IF(NEVTOP.EQ.2)IL=IEVT(IC,IR)
C
C4-----IF THE CELL IS EXTERNAL IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 10
      C=EVTR(IC,IR)
      S=SURF(IC,IR)
      H=HNEW(IC,IR,IL)
C
C5-----IF AQUIFER HEAD IS GREATER THAN OR EQUAL TO SURF, ET IS CONSTANT
      IF(H.LT.S) GO TO 5
C
C5A-----SUBTRACT -EVTR FROM RHS
      RHS(IC,IR,IL)=RHS(IC,IR,IL) + C
      GO TO 10
C
C6-----IF DEPTH TO WATER>=EXTINCTION DEPTH THEN ET IS 0
      5 D=S-H
      X=EXDP(IC,IR)
      IF(D.GE.X)GO TO 10
C
C7-----LINEAR RANGE. ADD ET TERMS TO BOTH RHS AND HCOF.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)+C-C*S/X
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C/X
      10 CONTINUE
C
C8-----RETURN
      RETURN
      END

```

List of Variables for Module EVT1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
C	Module	Maximum ET rate.
D	Module	Depth to water.
EVTR	Package	DIMENSION (NCOL,NROW), Maximum ET rate.
EXDP	Package	DIMENSION (NCOL,NROW), Extinction depth.
H	Module	Head in the cell.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IEVT	Package	DIMENSION (NCOL,NROW), Layer number, for each horizontal-cell location, from which ET will be taken if the ET option (NEVTOP) is equal to two.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option. = 1, ET is from the top layer. = 2, ET at each horizontal cell location is from the layer-indicator array (IEVT).
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of finite-difference equation. RHS is an accumulation of terms from several different packages.
S	Module	ET surface elevation for a cell.
SURF	Package	DIMENSION (NCOL,NROW), Elevation of the ET surface.
X	Module	Extinction depth for a cell.

### Narrative for Module EVT1BD

This module calculates rates and volumes removed from the aquifer by evapotranspiration.

1. Clear the rate accumulator RATOUT.
2. If budget terms will be saved, clear the buffer (BUFF) in which they will be accumulated.
3. Process each horizontal-cell location one at a time calculating flow to evapotranspiration (STEPS 4-11).
4. Set the layer index (IL) equal to one.
5. If option 2 is in effect, get the layer index from the layer-indicator array (IEVT).
6. If the cell is external ( $IBOUND \leq 0$ ), bypass processing of the cell.
7. If the head in the aquifer is greater than the elevation of the ET surface, set the ET rate for the cell equal to the maximum ET rate. SKIP STEPS 8 AND 9.
8. If the depth to the water is greater than the extinction depth, bypass further processing of this cell. SKIP STEP 9.
9. Calculate the ET rate using the linear approximation.
10. Add the ET flow from the cell to the accumulator (RATOUT).
11. If the cell-by-cell flow terms are to be saved, add the ET rate to the buffer (BUFF).
12. If the cell-by-cell flow terms are to be saved, call module UBUDSV to write the buffer (BUFF) onto a disk.
13. Move RATOUT into the VBVL array for printing by BAS10T.
14. Add RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS10T.
15. Move the ET budget-term labels to VBNM for printing by BAS10T.
16. Increment the budget-term counter (MSUM).
17. RETURN.

## Flow Chart for Module EVT1BD

RATOUT is an accumulator to which all flows out of the aquifer are added.

BUFFER is an array in which values are stored as they are being gathered for printing or recording.

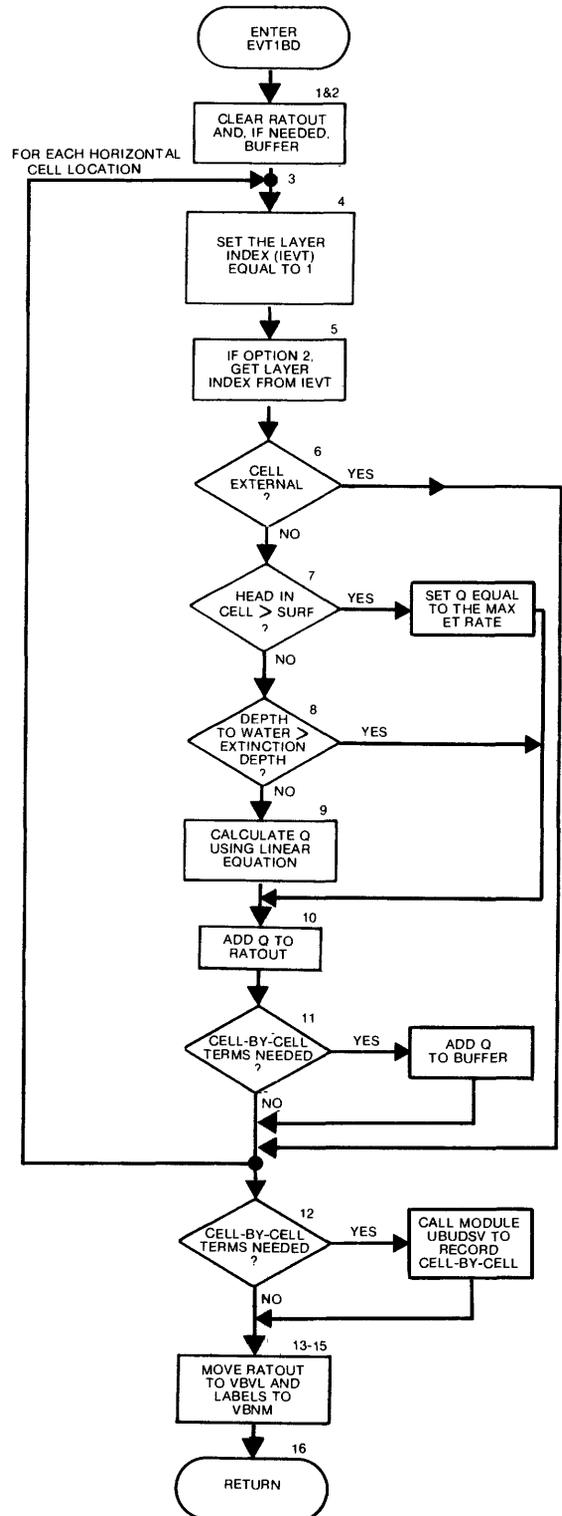
IEVT is an array containing a layer indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location from which ET is taken. It is used only if NEVTOP is equal to two.

SURF is an array containing the ET surface elevation for each horizontal cell location.

Q is the flow to ET from an individual cell.

VBVL is a table of budget entries calculated by component-of-flow packages for use in calculating the volumetric budget.

VBNM is a table of labels for budget terms.



```

SUBROUTINE EVT1BD(NEVTOP,IEVT,EVTR,EXDP,SURF,IBOUND,HNEW,
1          NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,KSTP,KPER,
2          IEVTCB,ICBCFL,BUFF,IOUT)
C-----VERSION 1405 10FEB1983 EVT1BD
C *****
C CALCULATE VOLUMETRIC BUDGET FOR EVAPOTRANSPIRATION
C *****
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
1 SURF(NCOL,NROW),IBOUND(NCOL,NROW,NLAY),
2 VBVL(4,20),VBNM(4,20),HNEW(NCOL,NROW,NLAY),
3 BUFF(NCOL,NROW,NLAY)
C DIMENSION TEXT(4)
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' ',' ',' ',' ET'/
C -----
C
C1-----CLEAR THE RATE ACCUMULATOR.
      RATOUT=0
C
C2-----IF CELL-BY-CELL FLOW TERMS WILL BE SAVED THEN CLEAR THE BUFFER.
      IBD=0
      IF(IEVTCB.LE.0 .OR. ICBCFL.EQ.0) GO TO 5
      IBD=1
      DO 4 IL=1,NLAY
      DO 4 IR=1,NROW
      DO 4 IC=1,NCOL
      BUFF(IC,IR,IL)=0.
4 CONTINUE
C
C3-----PROCESS EACH HORIZONTAL CELL LOCATION
      5 DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C4-----SET THE LAYER INDEX EQUAL TO 1
      IL=1
C
C5-----IF OPTION 2 IS SPECIFIED THEN GET LAYER INDEX FROM IEVT ARRAY
      IF(NEVTOP.EQ.2)IL=IEVT(IC,IR)
C
C6-----IF CELL IS EXTERNAL THEN IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 10
      C=EVTR(IC,IR)
      S=SURF(IC,IR)
      H=HNEW(IC,IR,IL)
C
C7-----IF AQUIFER HEAD => SURF,SET Q=MAX ET RATE
      IF(H.LT.S) GO TO 7
      Q=-C
      GO TO 9
C
C8-----IF DEPTH=>EXTINCTION DEPTH, ET IS 0

```

```

      7 X=EXDP(IC,IR)
        D=S-H
        IF(D.GE.X)GO TO 10
C
C9-----LINEAR RANGE . Q=-EVTR(H-EXEL)/EXDP
        Q=C*D/X-C
C
C10-----ACCUMULATE TOTAL FLOW RATE
        9 RATOUT=RATOUT-Q
C
C11-----IF CELL-BY-CELL FLOW TERMS TO BE SAVED THE ADD Q TO BUFFER.
        IF(IBD.EQ.1) BUFF(IC,IR,IL)=Q
        10 CONTINUE
C
C12-----IF C-B-C TO BE SAVED CALL MODULE UBUDSV TO RECORD THEM.
        IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IEVTCB,BUFF,NCOL,NROW,
          1                      NLAY,IOUT)
C
C13-----MOVE TOTAL ET RATE INTO VBVL FOR PRINTING BY BAS10T.
        VBVL(3,MSUM)=0.
        VBVL(4,MSUM)=RATOUT
C
C14-----ADD ET(ET RATE TIMES STEP LENGTH) TO VBVL
        VBVL(1,MSUM)=0.
        VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
C
C15-----MOVE BUDGET TERM LABELS TO VBNM FOR PRINT BY MODULE BAS10T
        VBNM(1,MSUM)=TEXT(1)
        VBNM(2,MSUM)=TEXT(2)
        VBNM(3,MSUM)=TEXT(3)
        VBNM(4,MSUM)=TEXT(4)
C
C16-----INCREMENT BUDGET TERM COUNTER
        MSUM=MSUM+1
C
C17-----RETURN
        RETURN
        END

```

List of Variables for Module EVT1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
C	Module	Maximum ET rate at a cell.
D	Module	Depth to water below the ET surface.
DELTA	Global	Length of the current time step.
EVTR	Package	DIMENSION (NCOL,NROW), Maximum ET rate.
EXDP	Package	DIMENSION (NCOL,NROW), Extinction depth.
H	Module	Head in the cell.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
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 for the current time step.
IEVT	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal-cell location from which ET will be taken if the ET option (NEVTOP) is equal to two.
IEVTCB	Package	Flag. If IEVTCB > 0 and ICBCFL ≠ 0, cell-by-cell flow terms for the EVT1 Package will be recorded on UNIT = IEVTCB.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option. = 1, ET is from the top layer. = 2, ET at each horizontal-cell location is from the layer-indicator array (IEVT).
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
Q	Module	Flow from ET into the cell. (Reverse the sign to get the flow to ET.)

List of Variables for Module EVT1BD (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
RATOUT	Module	Accumulator for the total flow out of the flow field to ET.
S	Module	Elevation of the ET surface for a cell.
SURF	Package	DIMENSION (NCOL,NROW), Elevation of the ET surface.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the 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.
X	Module	Extinction depth for a cell.

## CHAPTER 11

### GENERAL-HEAD BOUNDARY PACKAGE

#### Conceptualization and Implementation

A general-head boundary (GHB) consists of a source of water outside the modeled area which supplies water to a cell in the modeled area at a rate proportional to the head difference between the source and the cell. The rate at which water is supplied to cell  $i,j,k$  is given by the expression

$$Q_{i,j,k,m} = C_m(HB_m - h_{i,j,k}) \quad (78)$$

where

$Q_{i,j,k,m}$  is the rate at which water is supplied to the cell from boundary  $m$  ( $L^3t^{-1}$ );

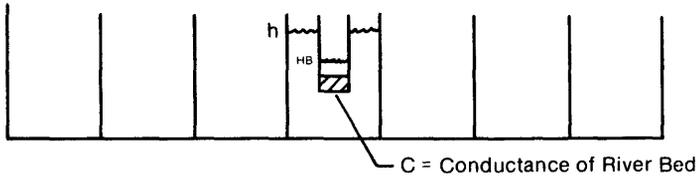
$C_m$  is the constant of proportionality for boundary  $m$  ( $L^2t^{-1}$ );

$HB_m$  is the head at the source boundary  $m$  (L); and

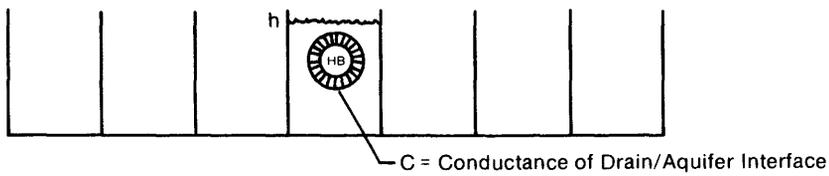
$h_{i,j,k}$  is the head in the cell (L).

The source of water could be a gaining river in which case the constant of proportionality is the conductance of the riverbed (fig. 44(a)). The source could be a buried drain (i.e., a negative source); then the constant is a function of the material around the drain and the size and spacing of openings in the skin of the drain (fig. 44(b)). It could also be the head in the aquifer outside the simulated area in which case the constant of proportionality is the hydraulic conductance of the material between the known head and boundary of the simulated area (fig. 44(c)). Although the GHB Package can be used to simulate the situations shown in figure 44, it should be done with great care. In the first two cases, the GHB Package, which deals with a single linear relationship (fig. 45), is more restrictive

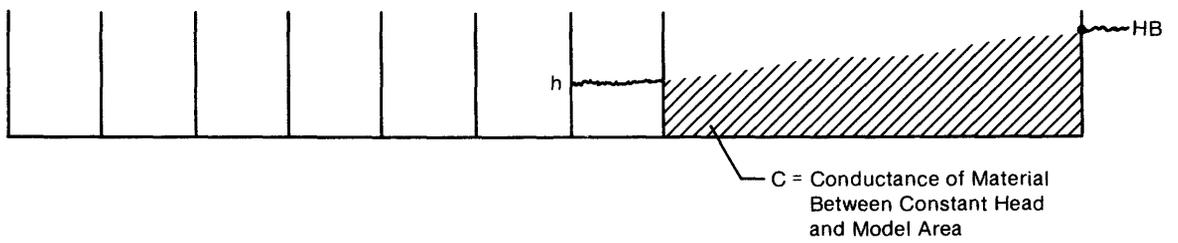
$$Q = C (HB-h)$$



A = Gaining Stream

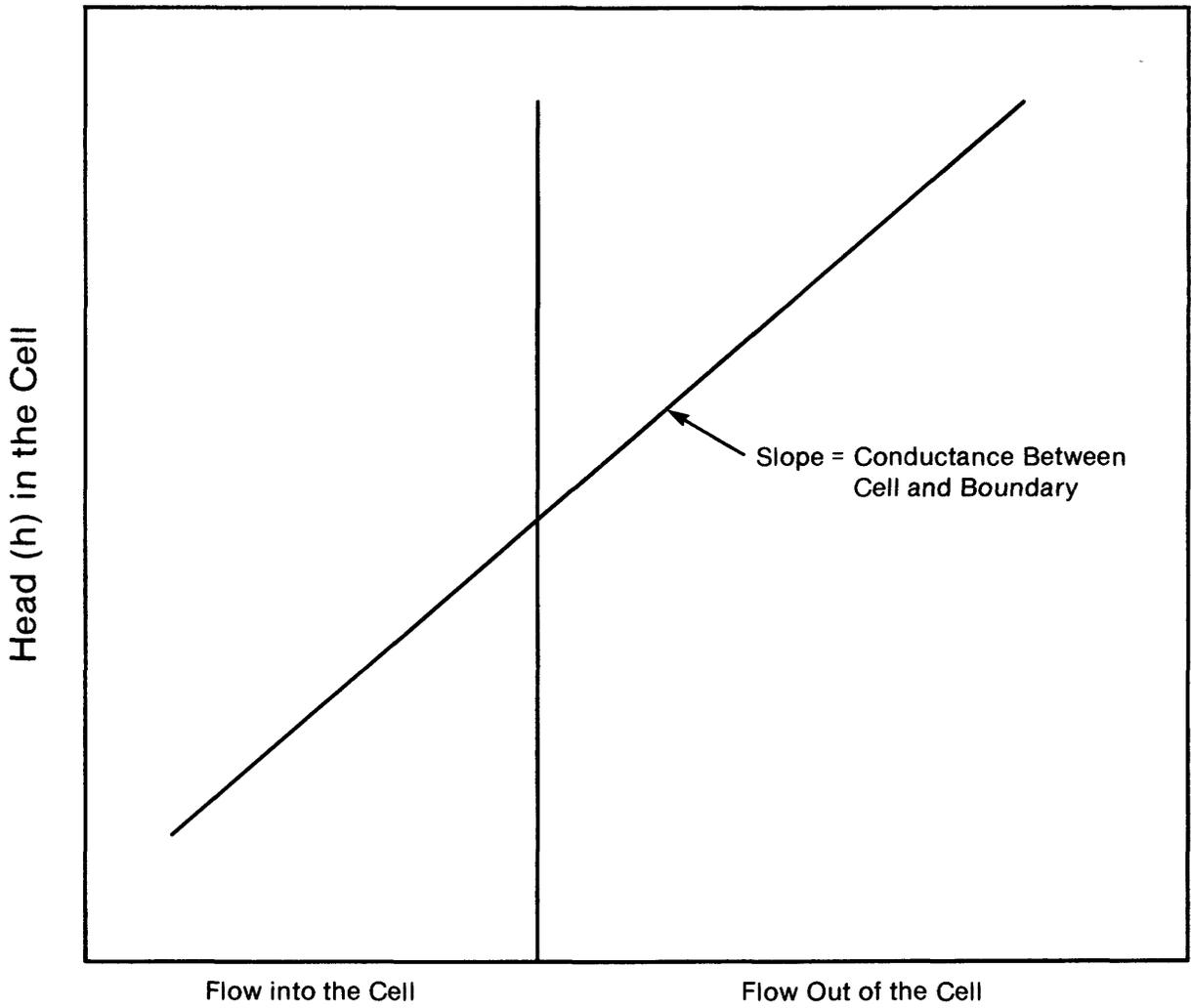


B = Buried Drain



C = Constant Head Outside the Model Area

Figure 44.—Three situations which can be simulated using the General Head Boundary Package: (a) a gaining stream, (b) a buried drain, and (c) horizontal leakage.



Flow to a General Head Boundary

Figure 45.—Flow from a general-head boundary as a function of head in the aquifer.

than the River or Drain Packages, each of which deal with two linear relationships. In the third case, the GHB Package does not attempt to account for change in storage in the aquifer material between the boundary head and the simulated area.

Data describing each GHB, which is stored in a list, is specified by the user for each stress period. Input for each boundary consists of the location of the boundary cell--layer, row, and column--the boundary head, and the constant of proportionality. During the formulation phase of each iteration, the term  $-C*HB$  is added to the accumulator RHS and the term  $-C$  is added to the accumulator HCOF.

## General-Head Boundary Package Input

Input for the General-Head Boundary (GHB) Package is read from the unit specified in IUNIT(7).

FOR EACH SIMULATION

GHB1AL

1. Data:    MXBND        IGHBCB  
Format:    I10         I10

FOR EACH STRESS PERIOD

GHB1RP

2. Data:    ITMP  
Format:    I10

3. Data:	Layer	Row	Column	Boundary Head	Cond
Format:	I10	I10	I10	F10.0	F10.0

(Input item 3 normally consists of one record for each GHB.  
If ITMP is negative or zero, item 3 is not read.)

### Explanation of Fields Used in Input Instructions

MXBND--is the maximum number of general-head boundary cells at one time.

IGHBCB--is a flag and a unit number.

If IGHBCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL (see Output Control) is set.

If IGHBCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IGHBCB < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

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

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

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

Boundary head--is the head on the boundary.

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

SAMPLE INPUT TO THE GENERAL HEAD BOUNDARY PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS			
1	{MXBND, IGHBCB}	6	24		
2	STRESS PERIOD 1 {ITMP}	4			
3	FIRST BOUNDARY {Layer, Row, Column, Head, Conductance	2	5	6	235.0
3	SECOND BOUNDARY {Layer, Row, Column, Head, Conductance	2	4	6	230.0
3	THIRD BOUNDARY {Layer, Row, Column, Head, Conductance	2	5	8	250.0
3	FOURTH BOUNDARY {Layer, Row, Column, Head, Conductance	2	7	6	235.0
2	STRESS PERIOD 2 {ITMP}	-1			
2	STRESS PERIOD 3 {ITMP}	-1			
2	STRESS PERIOD 4 {ITMP}	6			
3	FIRST BOUNDARY {Layer, Row, Column, Head, Conductance	2	5	6	235.0
3	SECOND BOUNDARY {Layer, Row, Column, Head, Conductance	2	4	6	230.0
3	THIRD BOUNDARY {Layer, Row, Column, Head, Conductance	2	5	8	250.0
3	FOURTH BOUNDARY {Layer, Row, Column, Head, Conductance	2	7	6	235.0
3	FIFTH BOUNDARY {Layer, Row, Column, Head, Conductance	2	9	6	235.0
3	SIXTH BOUNDARY {Layer, Row, Column, Head, Conductance	2	10	6	250.0

## Module Documentation for the General-Head Boundary Package

The General-Head Boundary Package (GHB1) consists of four modules, all of which are called by the MAIN program. The modules are:

- GHB1AL      Allocates space for an array that contains  
                 the general-head boundary list (BNDS).
  
- GHB1RP      Reads location, boundary head, and boundary  
                 conductance ( $C_m$ ) of each cell containing  
                 general-head boundary  $m$ .
  
- GHB1FM      Adds the terms  $-C_m$  and  $-C_m H B_m$  to the accumulators  
                  $HCOF_{i,j,k}$  and  $RHS_{i,j,k}$ , respectively.
  
- GHB1BD      Calculates the rates and accumulated volume of  
                 flow to and from general-head boundaries.

### Narrative for Module GHBIAL

This module allocates space in the X array to store the list of general-head boundaries (GHB).

1. Print a message identifying the package and initialize NBOUND (number of general-head boundaries).
2. Read and print MXBND (the maximum number of general-head boundaries) and IGHBCB (the unit number for saving cell-by-cell flow terms or a flag indicating that cell-by-cell flow terms should be printed).
3. Set LCBNDS, which will point to the first element in the boundary list (BNDS), equal to ISUM which is currently pointing to the first unallocated element in the X array.
4. Calculate the amount of space needed for the boundary list (five values for each boundary--row, column, layer, head, and conductance) and add it to ISUM so that it continues to point to the first unallocated element in X.
5. Print the number of elements in the X array used by the GHB Package.
6. RETURN.

## Flow Chart for Module GHB1AL

NBOUND is the number of general-head boundaries being simulated at any given time.

MXBND is the maximum number of general-head boundaries simulated.

IGHBCB is a flag and a unit number.

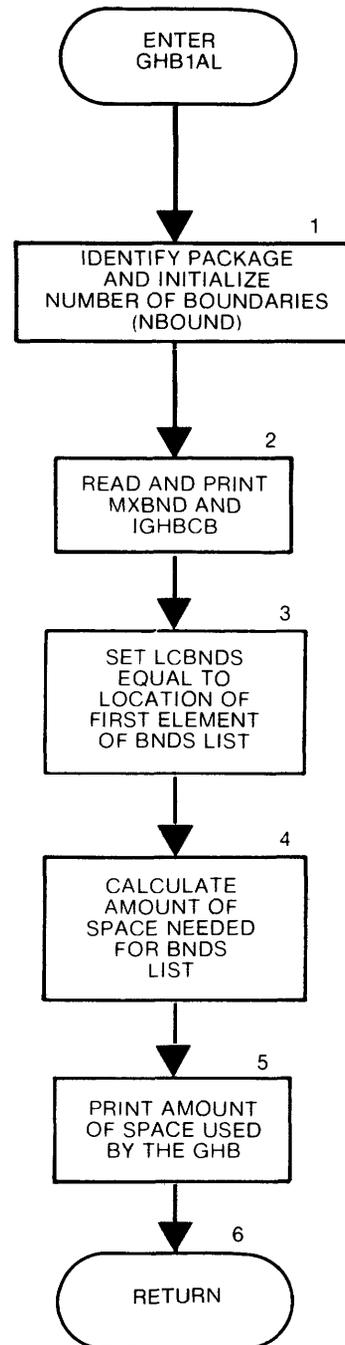
If  $IGHBCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If  $IGHBCB = 0$ , cell-by-cell flow terms will not be printed or recorded.

If  $IGHBCB < 0$ , the boundary leakage for each cell will be printed whenever ICBCFL is set.

LCBNDS is the location in the X array of the list of general-head boundaries data (BNDS).

BNDS is a table containing data for general-head boundaries.



```

SUBROUTINE GHBIAL(ISUM,LENX,LCBNDS,NBOUND,MXBND,IN,IOUT,
1          IGHBCB)
C
C-----VERSION 0940 08DEC1983 GHBIAL
C *****
C ALLOCATE ARRAY STORAGE FOR HEAD-DEPENDENT BOUNDARIES
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----IDENTIFY PACKAGE AND INITIALIZE # OF GENERAL HEAD BOUNDS
WRITE(IOUT,1)IN
1 FORMAT(1H0,'GHBI -- GHB PACKAGE, VERSION 1, 12/08/83',
2' INPUT READ FROM UNIT',I3)
NBOUND=0
C
C2-----READ AND PRINT MXBND AND IGHBCB (MAX # OF BOUNDS AND UNIT
C2-----FOR CELL-BY-CELL FLOW TERMS FOR GHB)
READ(IN,2) MXBND,IGHBCB
2 FORMAT(2I10)
WRITE(IOUT,3) MXBND
3 FORMAT(1H , 'MAXIMUM OF',I5,' HEAD-DEPENDENT BOUNDARY NODES')
IF(IGHBCB.GT.0) WRITE(IOUT,9) IGHBCB
9 FORMAT(1X,'CELL-BY-CELL FLOW WILL BE RECORDED ON UNIT',I3)
IF(IGHBCB.LT.0) WRITE(IOUT,8)
8 FORMAT(1X,'CELL-BY-CELL FLOW WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3-----SET LCBNDS EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
LCBNDS=ISUM
C
C4-----CALCULATE AMOUNT OF SPACE USED BY THE GENERAL HEAD LIST.
ISP=5*MXBND
ISUM=ISUM+ISP
C
C5-----PRINT AMOUNT OF SPACE USED BY THE GHBI PACKAGE
WRITE(IOUT,4) ISP
4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR HEAD',
1' -DEPENDENT BOUNDARIES')
ISUM1=ISUM-1
WRITE(IOUT,5) ISUM1,LENX
5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X,' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6-----RETURN
RETURN
END

```

List of Variables for Module GHB1AL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IGHBCB	Package	Flag and a unit number. > 0, unit number on which the cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, boundary leakage for each cell will be printed whenever IGHBFLL is set.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISP	Module	Number of words in the X array allocated by this module.
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	ISUM-1.
LCBNDS	Package	Location in the X array of the first element of array BNDS.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current stress period.

### Narrative for Module GHB1RP

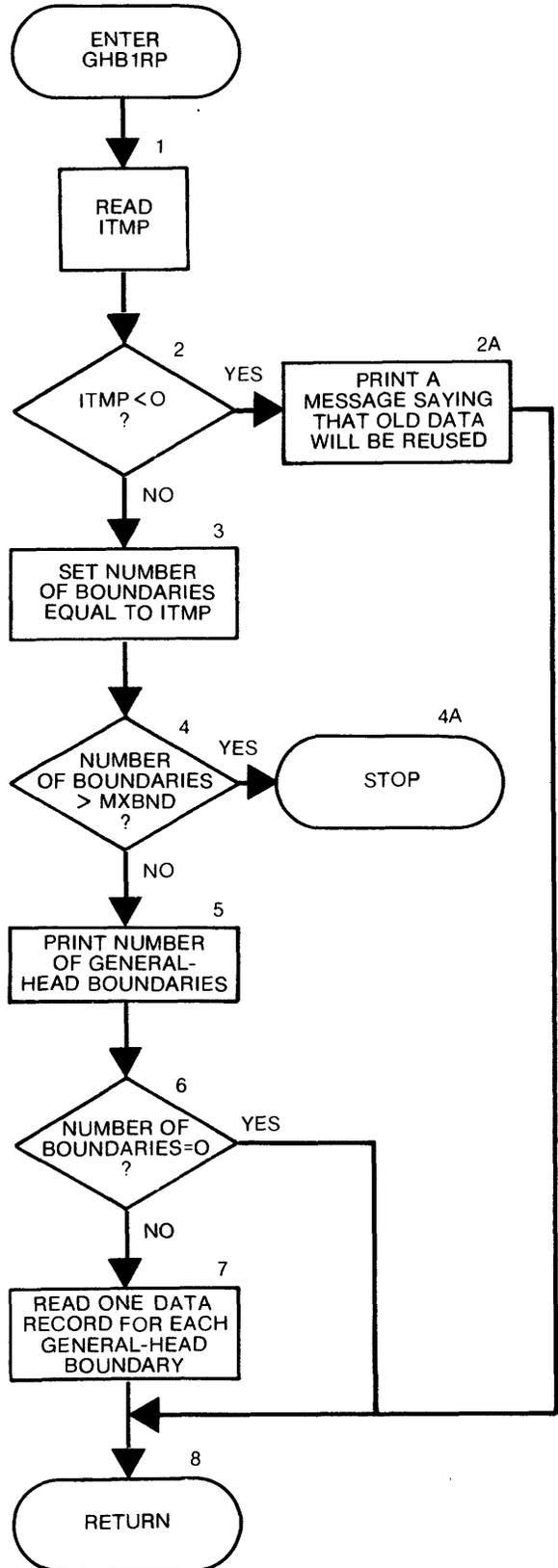
This module reads data to build the general-head boundary list.

1. Read ITMP. ITMP is the number of general-head boundaries or a flag indicating that data from the previous stress period should be reused.
2. Test ITMP. If ITMP is less than zero, the general-head boundary data read for the last stress period will be reused. Print a message to that effect and RETURN.
3. If ITMP is greater than or equal to zero, it is the number of general-head boundaries for this stress period. Set the number of general-head boundaries (NBOUND) in the current stress period equal to ITMP.
4. Compare the number of general-head boundaries (NBOUND) in the current stress period to the number specified as the maximum for the simulation (MXBND). If NBOUND is greater than MXBND, STOP.
5. Print the number of general-head boundaries in the current stress period (NBOUND).
6. See if there are any general-head boundaries. If there are none in the current stress period (NBOUND = 0), bypass further boundary processing (SKIP STEP 7).
7. Read and print the layer, row, column, head, and conductance for each general-head boundary.
8. RETURN.

Flow Chart for Module GHB1RP

ITMP is both a flag and a counter. If it is greater than or equal to zero, it is the number of general-head boundaries to be simulated during the stress period. If it is less than zero, it indicates that the boundaries simulated in the last stress period should be simulated in the current stress period.

MXBND is the maximum number of general-head boundaries to be simulated.



```

SUBROUTINE GH81RP(BNDS,NBOUND,MXBND,IN,IOUT)
C
C
C-----VERSION 1651 02FEB1983 GH81RP
C *****
C READ DATA FOR GH8
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION BNDS(5,MXBND)
C -----
C
C1-----READ ITMP(# OF GENERAL HEAD BOUNDS OR FLAG TO REUSE DATA.)
      READ(IN,8) ITMP
      8 FORMAT(I10)
C
C2-----TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD
      WRITE(IOUT,7)
      7 FORMAT(1H0,'REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS',
      1 ' PERIOD')
      GO TO 260
C
C3-----IF ITMP=>0 THEN IT IS THE # OF GENERAL HEAD BOUNDS.
      50 NBOUND=ITMP
C
C4-----IF MAX NUMBER OF BOUNDS IS EXCEEDED THEN STOP
      IF(NBOUND.LE.MXBND) GO TO 100
      WRITE(IOUT,99) NBOUND,MXBND
      99 FORMAT(1H0,'NBOUND(',I4,') IS GREATER THAN MXBND(',I4,')')
C
C4A-----ABNORMAL STOP
      STOP
C
C5-----PRINT # OF GENERAL HEAD BOUNDS THIS STRESS PERIOD
      100 WRITE(IOUT,1) NBOUND
      1 FORMAT(1H0,'/1X,15,' HEAD-DEPENDENT BOUNDARY NODES')
C
C6-----IF THERE ARE NO GENERAL HEAD BOUNDS THEN RETURN.
      IF(NBOUND.EQ.0) GO TO 260
C
C7-----READ & PRINT DATA FOR EACH GENERAL HEAD BOUNDARY.
      WRITE(IOUT,3)
      3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X
      1,'COL ELEVATION CONDUCTANCE BOUND NO. '/1X,15X,60('-'))
      DO 250 II=1,NBOUND
      READ (IN,4) K,I,J,BNDS(4,II),BNDS(5,II)
      4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,BNDS(4,II),BNDS(5,II),II
      5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,I8)
      BNDS(1,II)=K
      BNDS(2,II)=I
      BNDS(3,II)=J
      250 CONTINUE
C
C8-----RETURN
      260 RETURN
      END

```

List of Variables for Module GHBI RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BNDS	Package	DIMENSION (5, MXBND), Layer, row, column, head and conductance from boundary for each general-head boundary.
I	Module	Row number.
II	Module	Index for general-head boundaries.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of boundaries. $\geq 0$ , number of bounds active during the current stress period. $< 0$ , same bounds active during the last stress period will be active during the current stress period.
J	Module	Column number.
K	Module	Layer number.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current stress period.

### Narrative for Module GHB1FM

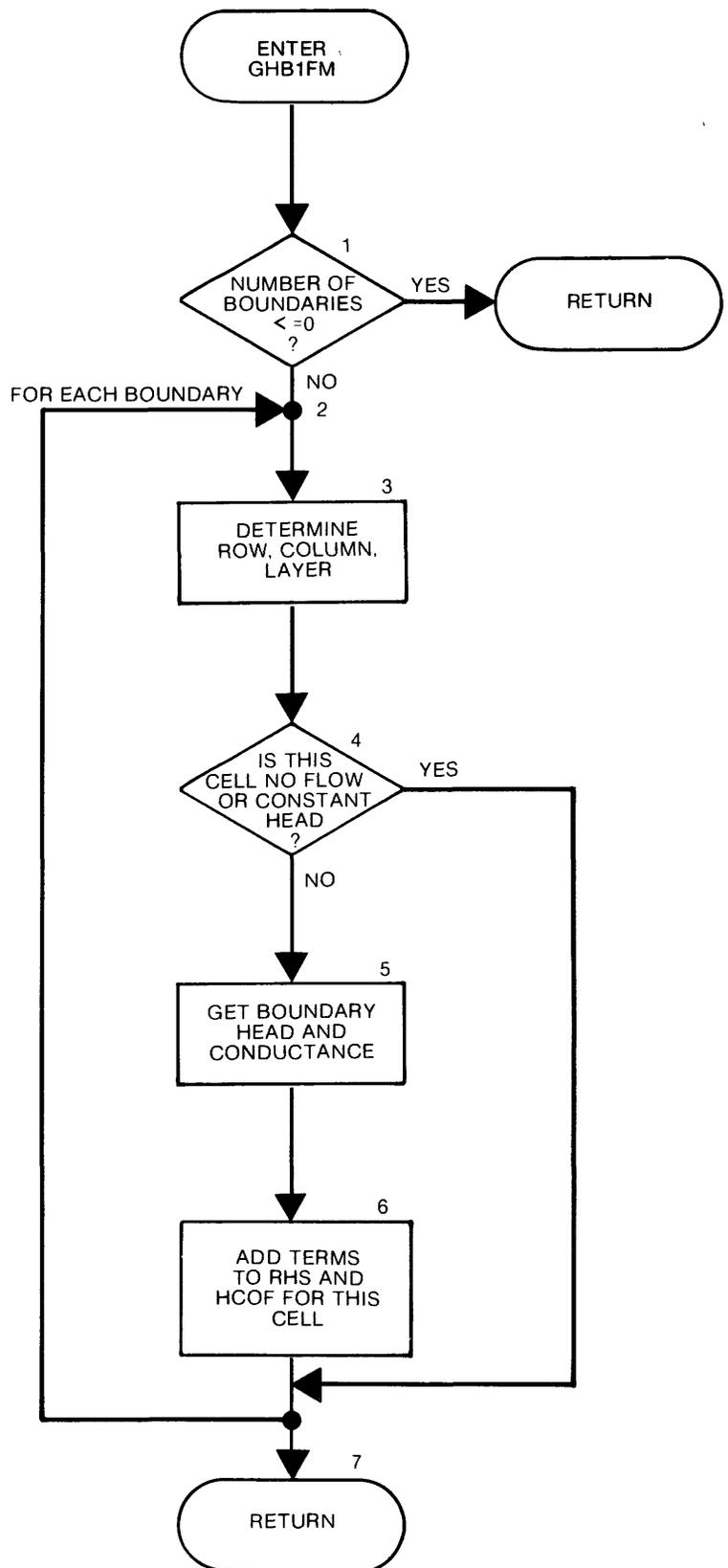
This module adds terms representing riverhead boundaries to the accumulators HCOF and RHS.

1. If NBOUND is less than or equal to zero in the current stress period, there are no general-head boundaries. RETURN.
2. For each boundary in the BNDS list, DO STEPS 3-6.
3. Determine the column (IC), row (IR), and layer (IL).
4. If the cell is external ( $IBOUND(IC, IR, IL) \leq 0$ ), bypass processing on this boundary and go on to the next one.
5. If the cell is internal, get the boundary data (head and conductance).
6. Add the  $-C*HB$  term (C is the conductance and HB is the boundary head) to the accumulator RHS and the term  $-C$  to the accumulator HCOF.
7. RETURN.

# Flow Chart for Module GHB1FM

RHS is an accumulator in which the right hand side of the equation is formulated.

HCOF is an accumulator in which the coefficient of head in the cell is formulated.



```

      SUBROUTINE GHBFM(NBOUND,MXBND,BNDS,HCOF,RHS,IBOUND,
1          NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1605 02FEB1983 GHBFM
C *****
C ADD GHF TERMS TO RHS AND HCOF
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION BNDS(5,MXBND),HCOF(NCOL,NROW,NLAY),
1     RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C -----
C
C1-----IF NBOUND<=0 THEN THERE ARE NO GENERAL HEAD BOUNDS. RETURN.
      IF(NBOUND.LE.0) RETURN
C
C2-----PROCESS EACH ENTRY IN THE GENERAL HEAD BOUND LIST (BNDS)
      DO 100 L=1,NBOUND
C
C3-----GET COLUMN, ROW AND LAYER OF CELL CONTAINING BOUNDARY
      IL=BNDS(1,L)
      IR=BNDS(2,L)
      IC=BNDS(3,L)
C
C4-----IF THE CELL IS EXTERNAL THEN SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C5-----SINCE THE CELL IS INTERNAL GET THE BOUNDARY DATA.
      HB=BNDS(4,L)
      C=BNDS(5,L)
C
C6-----ADD TERMS TO RHS AND HCOF
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-C*HB
100 CONTINUE
C
C7-----RETURN
      RETURN
      END

```

List of Variables for Module GHB1FM

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BNDS	Package	DIMENSION (5,MXBND), Layer, row, column, head and conductance from boundary for each general-head boundary.
C	Module	Conductance from the external boundary.
HB	Module	Head on boundary.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
L	Module	Index for boundaries.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current 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.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

## Narrative for Module GHB1BD

This module calculates rates and volumes transferred between the aquifer and general-head boundaries.

1. Initialize the cell-by-cell flow-term flag (IBD) and the rate accumulator (RATOUT).
2. If there are no general-head boundaries (NBOUND = 0), skip down to step 13 and put zeros into the budget terms for general-head boundaries.
3. Test to see if cell-by-cell flow terms are to be saved on disk. They will not be saved if either of the following conditions hold: (1) this is not the proper time step (ICBCFL = 0) or (2) cell-by-cell flow terms are not needed for general-head boundaries during this simulation (IGHBCB  $\leq$  0). If cell-by-cell flow terms will be saved for this package, clear the buffer in which they will be accumulated (BUFF) and set the cell-by-cell flow-term flag (IBD).
4. For each general-head boundary, DO STEPS 5-13 accumulating flows from or into the general-head boundary.
5. Determine the row, column, and layer of the cell containing the general-head boundary.
6. If the cell is external (IBOUND(I,J,K)  $\leq$  0), bypass further processing of this boundary.
7. Get the boundary parameters from the boundary list (BNDS).
8. Set RATE equal to the boundary conductance times the boundary head minus the head in the cell (RATE = C\*(HB - HHNEW)).

9. If cell-by-cell flow terms are to be printed ( $IGHBCB < 0$  and  $ICBCFL \neq 0$ ), print RATE.

10. If budget terms for individual cells are to be saved, add the RATE to the buffer (BUFF).

11. Check to see whether flow is into or out of the aquifer.

12. If RATE is negative, add it to RATOUT.

13. If RATE is positive, add it to RATIN.

14. See if cell-by-cell flow terms for individual cells are to be saved ( $IBD = 1$ ). If they are, call module UBUDSV to record the buffer (BUFF) onto disk.

15. Move RATIN and RATOUT into the VBVL array for printing by BAS10T. Add RATIN and RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS10T. Move the general-head boundary budget-term labels to VBNM for printing by BAS10T.

16. Increment the budget-term counter (MSUM). See the section in the Basic Package for a detailed explanation of VBVL, VBNM, and MSUM.

17. RETURN.

## Flow Chart for Module GHB1BD

IBD is a flag which, if set, causes cell-by-cell flow terms for general-head boundary to be recorded.

EXTERNAL: a cell is said to be external if it is either no flow or constant head (i.e., an equation is not formulated for the cell).

RATE is the leakage rate into the aquifer from the boundary in a cell.

BUFFER is an array in which values are stored as they are being gathered for printing or recording.

RATOUT is an accumulator to which all flows out of the aquifer are added.

RATIN is an accumulator to which all flows into the aquifer are added.

C is the conductance between the boundary and the cell.

HB is the boundary head.

HHNEW is the head in the cell.

IGHBCB is a flag and a unit number.

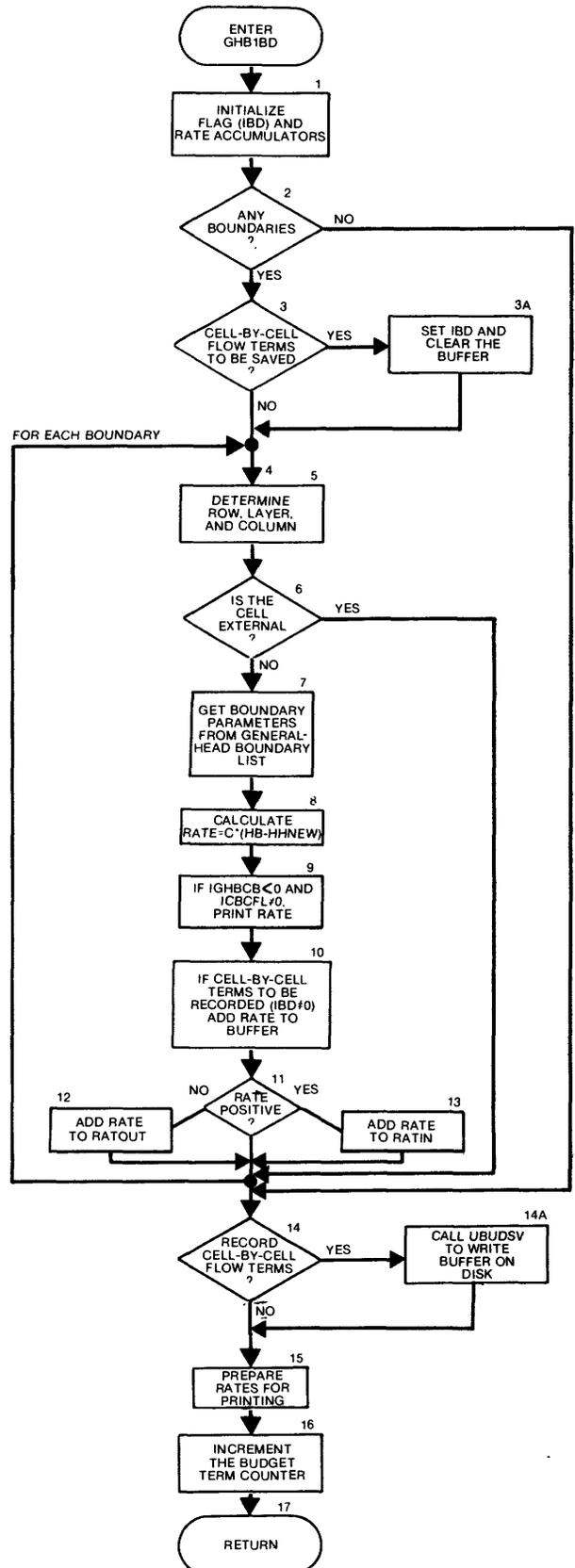
If  $IGHBCB > 0$ , it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

If  $IGHBCB = 0$ , cell-by-cell flow terms will not be printed or recorded.

If  $IGHBCB < 0$ , boundary leakage for each cell will be printed whenever ICBCFL is set.

ICBCFL is a flag.

If  $ICBCFL \neq 0$ , cell-by-cell flow terms will be either recorded or printed depending on IGHBCB for the current time step.



```

SUBROUTINE GHBI BD(NBOUND, MXBND, VBNM, VBVL, MSUM, BNDS, DELT, HNEW,
1  NCOL, NROW, NLAY, IBOUND, KSTP, KPER, IGHBCB, ICBCFL, BUFF, IOUT)
C
C-----VERSION 1152 20MAY1983 GHBI BD
C *****
C CALCULATE VOLUMETRIC BUDGET FOR GHBI
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW
C DIMENSION VBNM(4,MSUM),VBVL(4,MSUM),BNDS(5,MXBND),
1 HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
2 BUFF(NCOL,NROW,NLAY)
C DIMENSION TEXT(4)
C DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' HEA','D DE','P BO','UNDS'/
C -----
C
C1-----INITIALIZE CELL-BY-CELL FLOW TERM FLAG (IBD) AND
C1-----ACCUMULATORS (RATIN AND RATOUT)
C IBD=0
C RATOUT=0.
C RATIN=0.
C
C2-----IF NO BOUNDARIES THEN KEEP ZEROES IN ACCUMULATORS.
C IF(NBOUND.EQ.0) GO TO 200
C
C3-----TEST TO SEE IF CELL-BY-CELL FLOW TERMS ARE NEEDED.
C IF(ICBCFL.EQ.0 .OR. IGHBCB.LE.0) GO TO 10
C
C3A-----SINCE CELL-BY-CELL FLOW TERMS ARE NEEDED CLEAR BUFFER & SET
C3A-----THE FLAG IBD.
C IBD=1
C DO 5 IL=1,NLAY
C DO 5 IR=1,NROW
C DO 5 IC=1,NCOL
C BUFF(IC,IR,IL)=0.
C 5 CONTINUE
C
C4-----FOR EACH GENERAL HEAD BOUND ACCUMULATE FLOW INTO AQUIFER
C 10 DO 100 L=1,NBOUND
C
C5-----GET LAYER, ROW AND COLUMN OF EACH GENERAL HEAD BOUNDARY.
C IL=BNDS(1,L)
C IR=BNDS(2,L)
C IC=BNDS(3,L)
C
C6-----IF CELL IS EXTERNAL THEN IGNORE IT.
C IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C7-----GET PARAMETERS FROM BOUNDARY LIST.
C HHNEW=HNEW(IC,IR,IL)
C HB=BNDS(4,L)
C C=BNDS(5,L)

```

```

C
C8-----CALCULATE THE FOW RATE INTO THE CELL
      RATE=C*(HB-HHNEW)
C
C9-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IGHBCB<0).
      IF(IGHBCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
1     KPER,KSTP,L,IL,IR,IC,RATE
900 FORMAT(1H0,4A4,' PERIOD',I3,' STEP',I3,' BOUNDARY',I4,
1     ' LAYER',I3,' ROW',I4,' COL',I4,' RATE',G15.7)
C
C10-----IF CELL-BY-CELL TERMS ARE TO BE SAVED THEN PUT RATE IN BUFFER
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+RATE
C
C11-----SEE IF FLOW IS INTO AQUIFER OR OUT OF AQUIFER.
      IF(RATE)94,100,96
C
C12-----FLOW IS OUT OF AQUIFER SUBTRACT RATE FROM RATOUT
      94 RATOUT=RATOUT-RATE
      GO TO 100
C
C13-----FLOW IS INTO AQUIFER ADD RATE TO RATIN
      96 RATIN=RATIN+RATE
      100 CONTINUE
C
C14-----IF CELL-BY-CELL TERMS ARE TO BE SAVED THEN CALL
C14-----UTILITY MODULE UBUDSV
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IGHBCB,BUFF,NCOL,NROW,
1     NLAY,IOUT)
C
C15-----MOVE RATES, VOLUMES AND LABELS INTO ARRAYS FOR PRINTING
200 VBVL(3,MSUM)=RATIN
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
      VBVL(4,MSUM)=RATOUT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C16-----INCREMENT THE BUDGET TERM COUNTER
      MSUM=MSUM+1
C
C17-----RETURN
      RETURN
      END

```

List of Variables for Module GH1BD

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BNDS	Package	DIMENSION (5, MXBND), Layer, row, column, head and conductance from the boundary for each general-head boundary.
BUFF	Global	DIMENSION (NCOL, NROW, NLAY), Buffer used to accumulate information before printing or recording it.
C	Module	Conductance from the external boundary.
DELT	Global	Length of the current time step.
HB	Module	Head on boundary.
HHNEW	Module	HNEW (J, I, K), Single precision.
HNEW	Global	DIMENSION (NCOL, NROW, NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBD	Package	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL, NROW, NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IGHBCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.
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 either printed or recorded (depending on IGHBCB) for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for general-head boundaries.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current 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.

List of Variables for Module GHBIID (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
RATE	Module	Flow from a bound into a cell. (Reverse the sign to get flow into the bound.)
RATIN	Module	Accumulator for the total flow into the flow field out of the bounds.
RATOUT	Module	Accumulator for the total flow out of the flow field into the bounds.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the 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.

## CHAPTER 12

### STRONGLY IMPLICIT PROCEDURE PACKAGE

#### Conceptualization and Implementation

The Strongly Implicit Procedure (SIP) is a method for iteratively solving a large system of simultaneous linear equations.

For cell  $i,j,k$ , the finite-difference equation was shown to be of the form

$$\begin{aligned}
 & CV_{i,j,k-1/2}h_{i,j,k-1} + CC_{i-1/2,j,k}h_{i-1,j,k} + CR_{i,j-1/2,k}h_{i,j-1,k} \\
 & + (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} \\
 & - CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h_{i,j,k} \\
 & + CR_{i,j+1/2,k}h_{i,j+1,k} + CC_{i+1/2,j,k}h_{i+1,j,k} \\
 & + CV_{i,j,k+1/2}h_{i,j,k+1} = RHS_{i,j,k}. \tag{79}
 \end{aligned}$$

One equation of this form is written for each cell in the finite-difference grid. It expresses the relationship among the heads at node  $i,j,k$  and at each of the six adjacent nodes at the end of a time step. Because the head at any node appears in the equation for that node and in the equation for the adjoining cells, the equations must be solved simultaneously. The solution of all of the equations consists of the head for each node. The notation used in equation 79 is used to illustrate the role of conductances between nodes. It is convenient to rewrite the equation in a notation that permits easier identification of coefficients to a given equation. In the new notation, based on that of Weinstein, Stone, and Kwan (1969), the developers of SIP, equation 79 can be written

$$\begin{aligned}
 & Z_{i,j,k}h_{i,j,k-1} + B_{i,j,k}h_{i-1,j,k} + D_{i,j,k}h_{i,j-1,k} + E_{i,j,k}h_{i,j,k} \\
 & + F_{i,j,k}h_{i,j+1,k} + H_{i,j,k}h_{i+1,j,k} + S_{i,j,k}h_{i,j,k+1} = Q_{i,j,k}. \tag{80}
 \end{aligned}$$

Notation can be simplified by writing the system of equations in matrix form

$$\overline{\overline{A}}\overline{h} = \overline{q}. \quad (81)$$

The two bars over the  $\overline{A}$  indicate that it is a square matrix. The single bars over the  $\overline{h}$  and the  $\overline{q}$  indicate that they are vectors. Figure 46 shows the elements of the coefficient matrix and the two vectors. Notice that there are very few nonzero elements (the matrix is sparse) and that they are on just seven diagonals (fig. 47).

The coefficients in equation 80 all have the index  $i,j,k$  to show that they belong to the equations for node  $i,j,k$ . Furthermore, the  $Z$  coefficient for the equation at node  $i,j,k$  ( $Z_{i,j,k}$ ), is equal to  $CL_{i,j,k-1/2}$  which is the same as the  $S$  coefficient for the equation at node  $i,j,k-1$  ( $S_{i,j,k-1}$ ), or more succinctly,

$$Z_{i,j,k} = S_{i,j,k-1}. \quad (82)$$

Similarly,

$$B_{i,j,k} = H_{i-1,j,k} \quad (83)$$

and

$$D_{i,j,k} = F_{i,j-1,k}. \quad (84)$$

Thus the coefficient matrix,  $\overline{\overline{A}}$ , is symmetric (fig. 48).

Direct methods for solution of systems of simultaneous equations factor the coefficient equation,  $\overline{\overline{A}}$ , into two matrices,  $\overline{L}$  and  $\overline{U}$ , such that in  $\overline{L}$ , all of the nonzero elements are on or below the main diagonal; and in  $\overline{U}$ , all of the nonzero elements are above the main diagonal (fig. 49). After factoring  $\overline{\overline{A}}$ , equation 81 can be solved using a method called "backward and forward



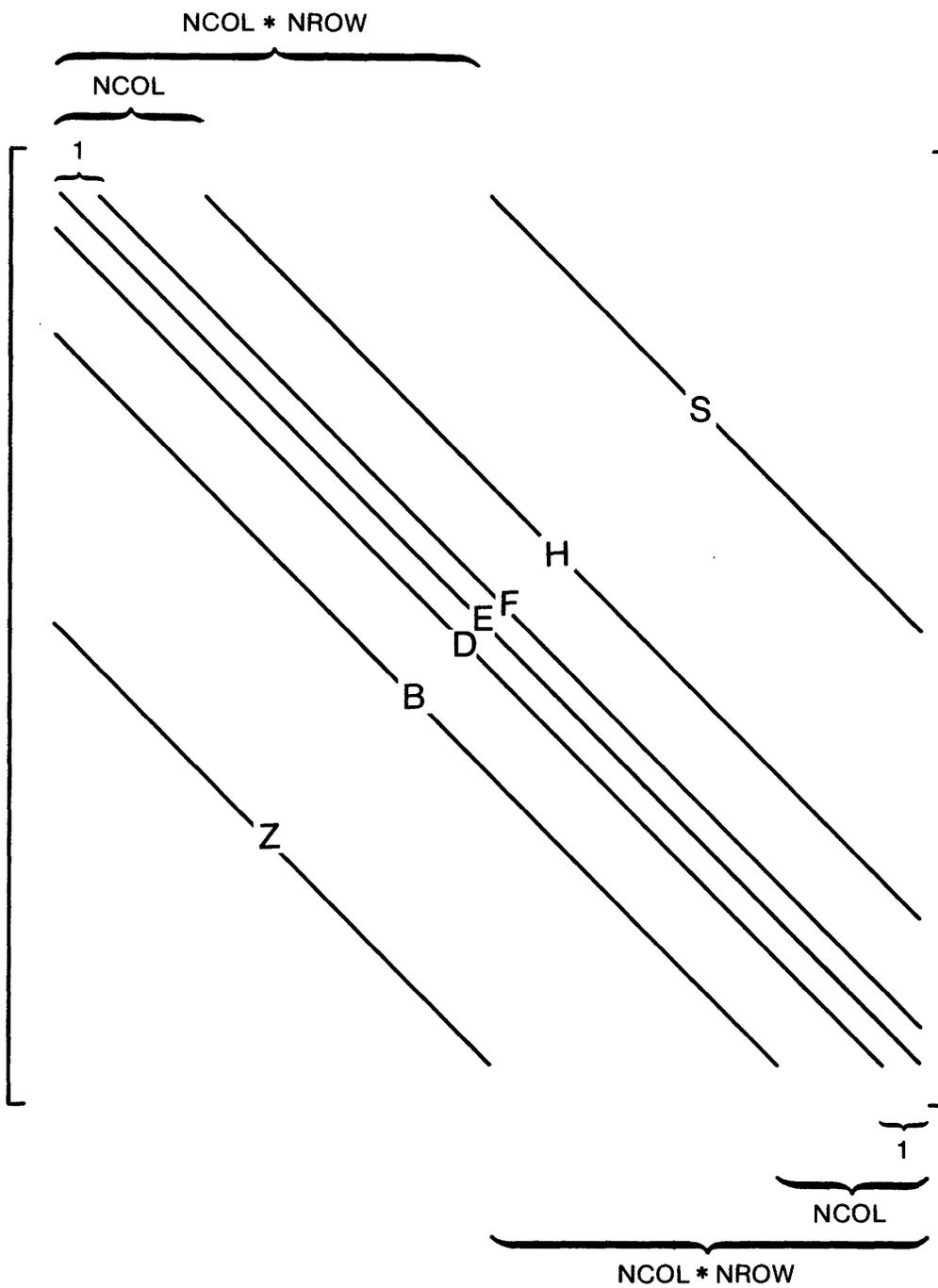


Figure 47.—Structure of coefficient matrix showing nonzero diagonals.



$$\begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 2 \\ 3 & 2 & -2 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 3 & 0 \\ 3 & -4 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}$$

Figure 49.—Decomposition of a coefficient matrix into lower and upper triangular matrices.

substitution." Unfortunately, even though  $\bar{A}$  is sparse,  $\bar{L}$  and  $\bar{U}$  are not sparse: a lot of computer memory and time is needed to calculate all of the nonzero elements in  $\bar{L}$  and  $\bar{U}$ . Furthermore, roundoff errors may become unacceptably large.

SIP tries to take advantage of the fact that the coefficient matrix is sparse. An attempt is made to find a matrix  $\bar{B}$  such that  $\overline{\bar{A}+\bar{B}}$  is "close" to  $\bar{A}$ , and  $\overline{\bar{A}+\bar{B}}$  can be factored easily into sparse matrices  $\bar{L}$  and  $\bar{U}$ . Once a matrix  $\bar{B}$  is constructed and  $\bar{B}\bar{h}$  is added to both sides of equation 81, the equation obtained is

$$(\overline{\bar{A}+\bar{B}})\bar{h} = \bar{q} + \bar{B}\bar{h}. \quad (85)$$

A vector  $\bar{h}$  which is a solution to equation 85 is also a solution to equation 81. If we knew the value of the right side of equation 85, we could easily solve the equation by factoring  $\overline{\bar{A}+\bar{B}}$  into  $\bar{L}$  and  $\bar{U}$  and using backward and forward substitution. Of course, we do not know the value  $\bar{B}\bar{h}$ . However, we could approximate it using our best estimate of  $\bar{h}$ . That would give rise to the iterative equation

$$(\overline{\bar{A}+\bar{B}})\bar{h}^m = \bar{q} + \bar{B}\bar{h}^{m-1}. \quad (86)$$

The vector  $\bar{h}^m$  is the m-th estimate of the vector  $\bar{h}$ . On the first iteration ( $m = 1$ ), the vector  $\bar{h}^{m-1}$  would be the heads at the start of the time step. On subsequent iterations,  $\bar{h}^{m-1}$  would be the head vector calculated at the previous iteration. The solution of equation 81 has been transformed to the problem of finding a matrix  $\bar{B}$  such that  $\overline{\bar{A}+\bar{B}}$  is close to  $\bar{A}$  and easily factored into lower and upper matrices  $\bar{L}$  and  $\bar{U}$ , where  $\bar{L}$  and  $\bar{U}$  are sparse.

It is convenient to specify that  $\bar{L}$  and  $\bar{U}$  will not only be sparse but that each will have only four nonzero diagonals (fig. 50). Multiplication of  $\bar{L}$  and  $\bar{U}$  gives the form of  $\overline{A+B}$ . There are six nonzero diagonals in  $\overline{A+B}$  that were not in  $\bar{A}$  (fig. 51). The relationship between elements in  $\bar{L}$  and  $\bar{U}$  and elements in  $\overline{A+B}$  are given by the following equations

$$Z'_{i,j,k} = a_{i,j,k} \quad (87-a)$$

$$A'_{i,j,k} = a_{i,j,k}e_{i,j,k-1} \quad (87-b)$$

$$T'_{i,j,k} = a_{i,j,k}f_{i,j,k-1} \quad (87-c)$$

$$B'_{i,j,k} = b_{i,j,k} \quad (87-d)$$

$$C'_{i,j,k} = e_{i-1,j,k}b_{i,j,k} \quad (87-e)$$

$$D'_{i,j,k} = c_{i,j,k} \quad (87-f)$$

$$E'_{i,j,k} = a_{i,j,k}g_{i,j,k-1} + b_{i,j,k}f_{i-1,j,k} + e_{i,j-1,k}c_{i,j,k} + d_{i,j,k} \quad (87-g)$$

$$F'_{i,j,k} = d_{i,j,k}e_{i,j,k} \quad (87-h)$$

$$G'_{i,j,k} = f_{i,j-1,k}c_{i,j,k} \quad (87-i)$$

$$H'_{i,j,k} = f_{i,j,k}d_{i,j,k} \quad (87-j)$$

$$U'_{i,j,k} = b_{i,j,k}g_{i-1,j,k} \quad (87-k)$$

$$R'_{i,j,k} = g_{i,j-1,k}c_{i,j,k} \quad (87-l)$$

$$S'_{i,j,k} = g_{i,j,k}d_{i,j,k} \quad (87-m)$$

If the subscript of an element in equations 87(a-m) places the element outside of the grid boundary, the element is assumed to be equal to zero. The 13 equations contain 20 unknown values, the elements of  $\bar{L}$ ,  $\bar{U}$ , and  $\overline{A+B}$ . Thus there are many matrices  $\bar{B}$  which can be added to  $\bar{A}$  so that the sum can be factored into upper and lower triangular matrices of the form of  $\bar{L}$  and  $\bar{U}$ .

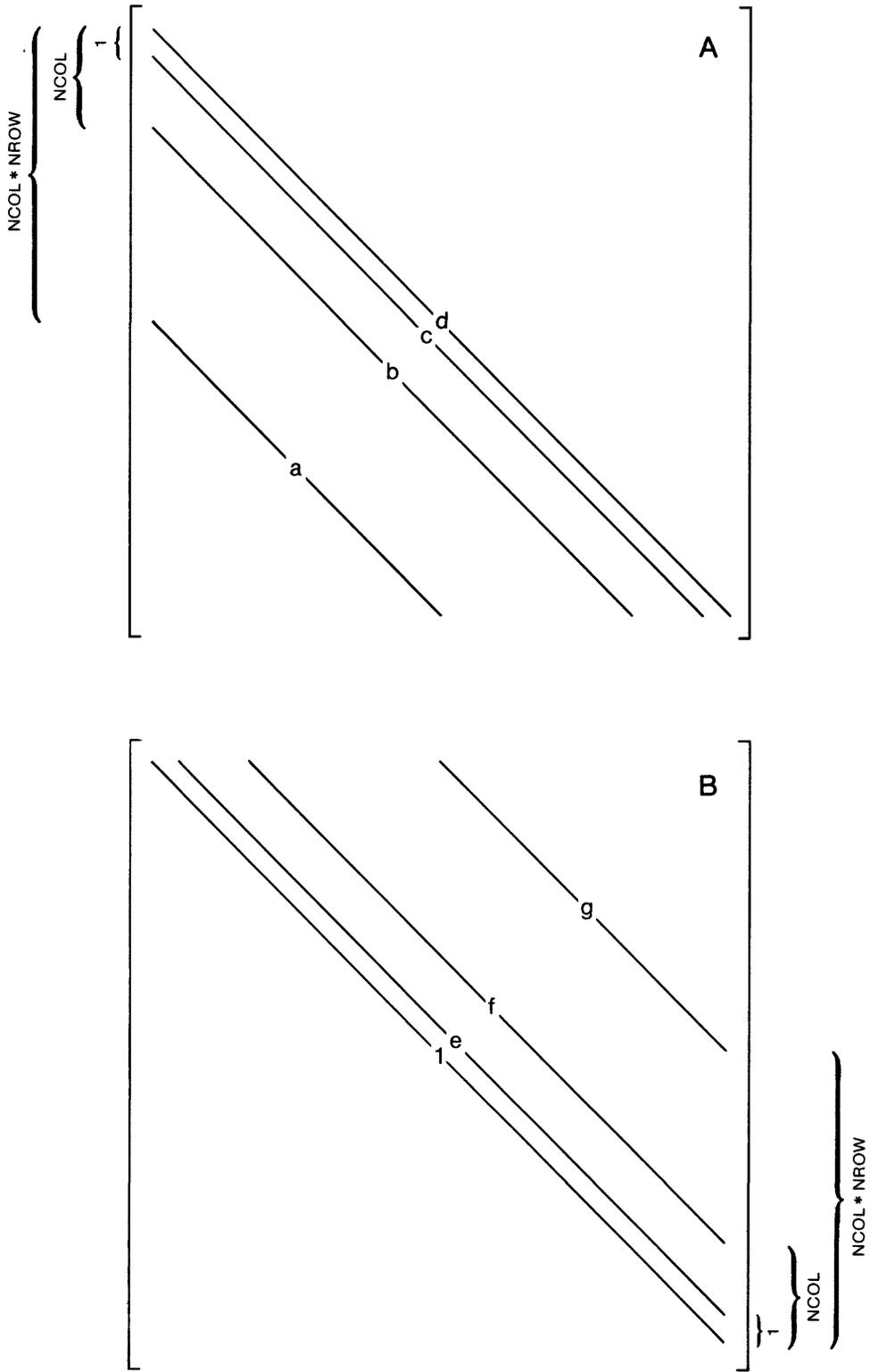


Figure 50.—Desired structure, showing nonzero diagonals, of  
 (a) the lower triangular factor  $\bar{L}$  of  $\bar{A+B}$ ,  
 and (b) the upper triangular factor  $\bar{U}$  of  $\bar{A+B}$ .

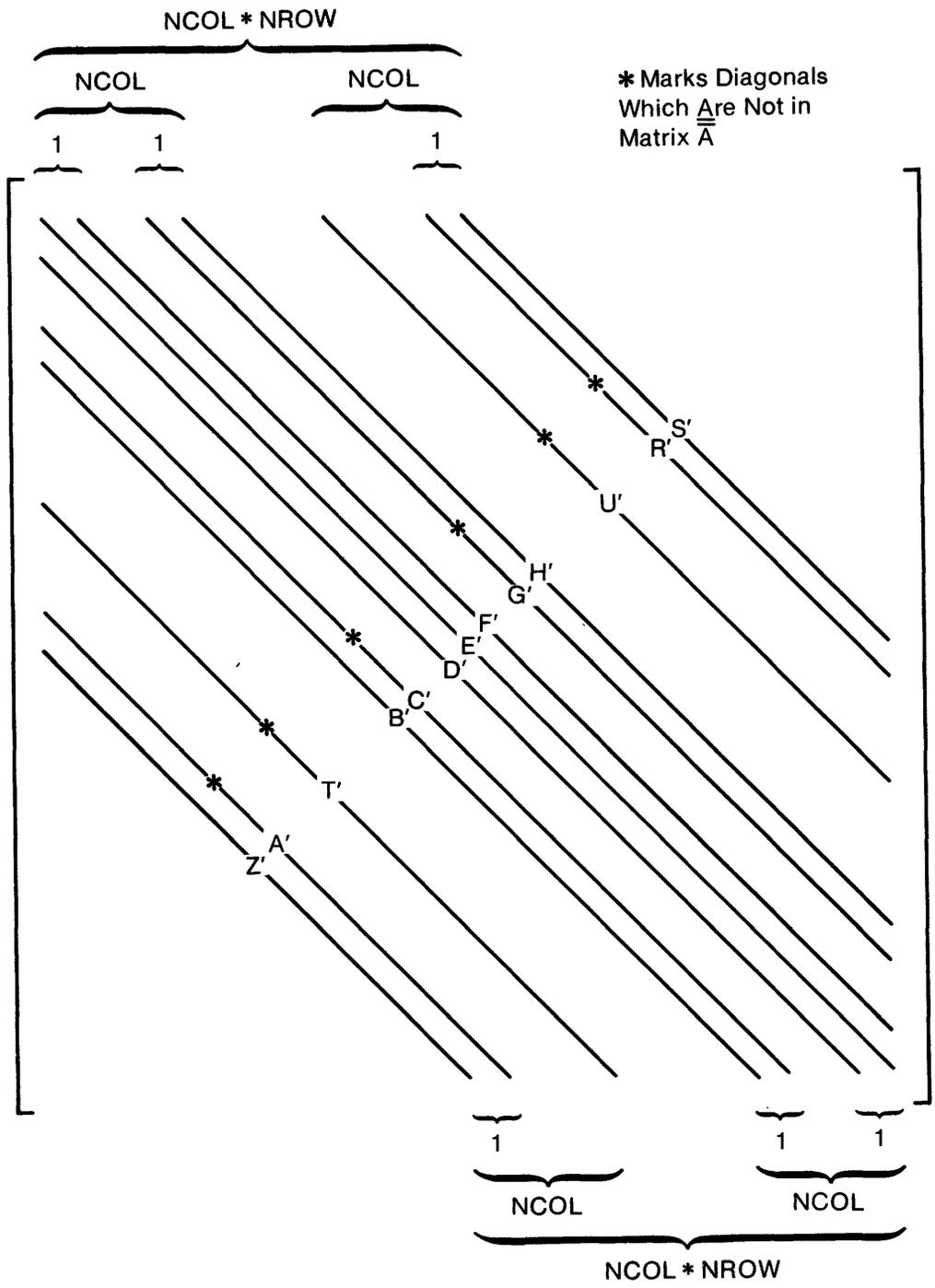


Figure 51.—Structure of matrix  $\overline{A+B}$  showing nonzero diagonals.

The requirement that  $\overline{A+B}$  is "close" to  $\overline{A}$  has not been used. The term "close" can be defined by the relation

$$\overline{A}h \approx (\overline{A+B})h. \quad (88)$$

In terms of a single equation in the system of equations,  $\overline{A+B}$  is close to  $\overline{A}$  means that

$$\begin{aligned} & Z'_{i,j,k}h_{i,j,k-1} + A'_{i,j,k}h_{i,j+1,k-1} + T'_{i,j,k}h_{i+1,j,k-1} \\ & + B'_{i,j,k}h_{i-1,j,k} + C'_{i,j,k}h_{i-1,j+1,k} + D'_{i,j,k}h_{i,j-1,k} \\ & + E'_{i,j,k}h_{i,j,k} + F'_{i,j,k}h_{i,j+1,k} + G'_{i,j,k}h_{i+1,j-1,k} \\ & + H'_{i,j,k}h_{i+1,j,k} + U'_{i,j,k}h_{i-1,j,k+1} + R'_{i,j,k}h_{i,j-1,k+1} \\ & + S'_{i,j,k}h_{i,j,k+1} \approx Z_{i,j,k}h_{i,j,k-1} + B_{i,j,k}h_{i-1,j,k} \\ & + D_{i,j,k}h_{i,j-1,k} + E_{i,j,k}h_{i,j,k} + F_{i,j,k}h_{i,j+1,k} + H_{i,j,k}h_{i+1,j,k} \\ & + S_{i,j,k}h_{i,j,k+1}. \end{aligned} \quad (89)$$

Equation 89 can be rearranged so that the extra terms are on the right side and the changes in the existing terms are on the left side.

$$\begin{aligned} & (Z_{i,j,k} - Z'_{i,j,k})h_{i,j,k-1} + (B_{i,j,k} - B'_{i,j,k})h_{i-1,j,k} \\ & + (D_{i,j,k} - D'_{i,j,k})h_{i,j-1,k} + (E_{i,j,k} - E'_{i,j,k})h_{i,j,k} \\ & + (F_{i,j,k} - F'_{i,j,k})h_{i,j+1,k} + (H_{i,j,k} - H'_{i,j,k})h_{i+1,j,k} \\ & + (S_{i,j,k} - S'_{i,j,k})h_{i,j,k+1} \approx A'_{i,j,k}h_{i,j+1,k-1} \\ & + T'_{i,j,k}h_{i+1,j,k-1} + C'_{i,j,k}h_{i-1,j+1,k} + G'_{i,j,k}h_{i+1,j-1,k} \\ & + U'_{i,j,k}h_{i-1,j,k+1} + R'_{i,j,k}h_{i,j-1,k+1} \end{aligned} \quad (90)$$

To partially cancel the effect of terms which contain head in cells not adjacent to cell  $i,j,k$ , the terms on the right side of equation 90 are multiplied by three "iteration parameters,"  $\alpha$ ,  $\beta$ , and  $\gamma$  between zero and one.

$$\begin{aligned}
& (Z_{i,j,k} - Z'_{i,j,k})h_{i,j,k-1} + (B_{i,j,k} - B'_{i,j,k})h_{i-1,j,k} \\
& + (D_{i,j,k} - D'_{i,j,k})h_{i,j-1,k} + (E_{i,j,k} - E'_{i,j,k})h_{i,j,k} \\
& + (F_{i,j,k} - F'_{i,j,k})h_{i,j+1,k} + (H_{i,j,k} - H'_{i,j,k})h_{i+1,j,k} \\
& + (S_{i,j,k} - S'_{i,j,k})h_{i,j,k+1} \approx \alpha^A_{i,j,k}h_{i,j+1,k-1} \\
& + \beta T'_{i,j,k}h_{i+1,j,k-1} + \gamma C'_{i,j,k}h_{i-1,j+1,k} + \gamma G'_{i,j,k}h_{i+1,j-1,k} \\
& + \beta U'_{i,j,k}h_{i-1,j,k+1} + \alpha R'_{i,j,k}h_{i,j-1,k+1} \tag{91}
\end{aligned}$$

The right side of equation 91 consists of terms which involve head in cells that are not adjacent to cell  $i,j,k$ . One such term is  $\alpha^A_{i,j,k}h_{i,j+1,k-1}$ . Node  $i,j+1,k-1$  is at a corner of a rectangle; the other three corners of which  $i,j,k-1$ ,  $i,j+1,k$ , and  $i,j,k$  are represented on the left side of equation 91. The head of one corner can be approximated by the sum of the heads at the adjacent corner minus the head at the opposite corner (fig. 52). Therefore,

$$h_{i,j+1,k-1} = h_{i,j+1,k} + h_{i,j,k-1} - h_{i,j,k} \tag{92a}$$

Similarly,

$$h_{i+1,j,k-1} = h_{i,j,k-1} + h_{i+1,j,k} - h_{i,j,k} \tag{92b}$$

$$h_{i-1,j+1,k} = h_{i-1,j,k} + h_{i,j+1,k} - h_{i,j,k} \tag{92c}$$

$$h_{i+1,j-1,k} = h_{i+1,j,k} + h_{i,j-1,k} - h_{i,j,k} \tag{92d}$$

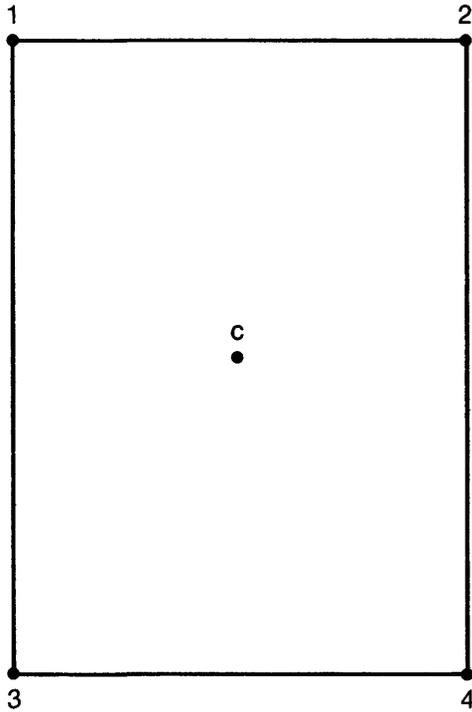
$$h_{i-1,j,k+1} = h_{i,j,k+1} + h_{i-1,j,k} - h_{i,j,k} \tag{92e}$$

$$h_{i,j-1,k+1} = h_{i,j,k+1} + h_{i,j-1,k} - h_{i,j,k} \tag{92f}$$

Substituting equations 92(a-f) into equation 91 and reorganizing gives

$$\begin{aligned}
& (Z'_{i,j,k} - Z_{i,j,k} + \alpha^A_{i,j,k} + \beta T'_{i,j,k})h_{i,j,k-1} \\
& + (B'_{i,j,k} - B_{i,j,k} + \gamma C'_{i,j,k} + \beta U'_{i,j,k})h_{i-1,j,k} \\
& + (D'_{i,j,k} - D_{i,j,k} + \gamma G'_{i,j,k} + \alpha R'_{i,j,k})h_{i,j-1,k} \\
& + (E'_{i,j,k} - E_{i,j,k} - \alpha^A_{i,j,k} - \beta T'_{i,j,k} - \gamma C'_{i,j,k} \\
& \quad - \gamma G'_{i,j,k} - \beta U'_{i,j,k} - \alpha R'_{i,j,k})h_{i,j,k} \\
& + (F'_{i,j,k} - F_{i,j,k} + \alpha^A_{i,j,k} + \gamma C'_{i,j,k})h_{i,j+1,k} \\
& + (H'_{i,j,k} - H_{i,j,k} + \beta T'_{i,j,k} + \gamma G'_{i,j,k})h_{i+1,j,k} \\
& + (S'_{i,j,k} - S_{i,j,k} + \beta U'_{i,j,k} + \alpha R'_{i,j,k})h_{i,j,k+1} \approx 0. \tag{93}
\end{aligned}$$

Suppose the Function  $f$  Is Known at 2, 3 and 4.



By Simple Interpolation the Function at the Center Can Be Approximated by

$$f_1(c) \approx \frac{f(2) + f(3)}{2}$$

and

$$f_2(c) \approx \frac{f(1) + f(4)}{2}$$

Suppose

$$f_1(c) \approx f_2(c)$$

Then

$$\frac{f(2) + f(3)}{2} \approx \frac{f(1) + f(4)}{2}$$

Therefore

$$f(1) \approx f(2) + f(3) - f(4)$$

Figure 52.—Estimation of a function at one corner of a rectangle in terms of the function at the other three corners.

The relation in equation 93 holds only if each coefficient is approximately equal to zero. Setting them equal to zero yields the equations

$$Z'_{i,j,k} - Z_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k} = 0 \quad (94a)$$

$$B'_{i,j,k} - B_{i,j,k} + \gamma C'_{i,j,k} + \beta U'_{i,j,k} = 0 \quad (94b)$$

$$D'_{i,j,k} - D_{i,j,k} + \gamma G'_{i,j,k} + \alpha R'_{i,j,k} = 0 \quad (94c)$$

$$\begin{aligned} E'_{i,j,k} - E_{i,j,k} - \alpha A'_{i,j,k} - \beta T'_{i,j,k} \\ - \gamma C'_{i,j,k} - \gamma G'_{i,j,k} \\ - \beta U'_{i,j,k} - \alpha R'_{i,j,k} = 0 \end{aligned} \quad (94d)$$

$$F'_{i,j,k} - F_{i,j,k} + \alpha A'_{i,j,k} + \gamma C'_{i,j,k} = 0 \quad (94e)$$

$$H'_{i,j,k} - H_{i,j,k} + \beta T'_{i,j,k} + \gamma G'_{i,j,k} = 0 \quad (94f)$$

$$S'_{i,j,k} - S_{i,j,k} + \beta U'_{i,j,k} + \alpha R'_{i,j,k} = 0 \quad (94g)$$

Equations 94(a-g) and 87(a-m) form a system of 20 equations in 20 unknowns which when solved, will yield the entries of  $\overline{A+B}$ ,  $\overline{L}$ , and  $\overline{U}$ . For example, substituting equations 87-a, b, and c into equation 94(a) and rearranging yields

$$a_{i,j,k} = Z_{i,j,k} / (1 + \alpha e_{i,j,k-1} + \beta f_{i,j,k-1}). \quad (95a)$$

Similarly,

$$b_{i,j,k} = B_{i,j,k} / (1 + \gamma e_{i-1,j,k} + \beta g_{i-1,j,k}) \quad (95b)$$

$$c_{i,j,k} = D_{i,j,k} / (1 + \gamma f_{i,j-1,k} + \alpha g_{i,j-1,k}) \quad (95c)$$

$$A'_{i,j,k} = a_{i,j,k} e_{i,j,k-1} \quad (95d)$$

$$C'_{i,j,k} = e_{i-1,j,k} b_{i,j,k} \quad (95e)$$

$$G'_{i,j,k} = f_{i,j-1,k} c_{i,j,k} \quad (95f)$$

$$R'_{i,j,k} = g_{i,j-1,k} c_{i,j,k} \quad (95g)$$

$$T'_{i,j,k} = a_{i,j,k} f_{i,j,k-1} \quad (95h)$$

$$U'_{i,j,k} = b_{i,j,k} g_{i-1,j,k} \quad (95i)$$

$$\begin{aligned} d_{i,j,k} = & E_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k} \\ & + \gamma C'_{i,j,k} + \gamma G'_{i,j,k} + \beta U'_{i,j,k} \\ & + \alpha R'_{i,j,k} - a_{i,j,k} g_{i,j,k-1} - b_{i,j,k} f_{i-1,j,k} \\ & - e_{i,j-1,k} c_{i,j,k} \end{aligned} \quad (95j)$$

$$e_{i,j,k} = (F_{i,j,k} - \alpha A'_{i,j,k} - \gamma C'_{i,j,k}) / d_{i,j,k} \quad (95k)$$

$$f_{i,j,k} = (H_{i,j,k} - \beta T'_{i,j,k} - \gamma G'_{i,j,k}) / d_{i,j,k} \quad (95l)$$

$$g_{i,j,k} = (S_{i,j,k} - \alpha R'_{i,j,k} - \beta U'_{i,j,k}) / d_{i,j,k} \quad (95m)$$

Recall that the goal was to find a matrix  $\bar{B}$  such that  $\overline{A+B}$  is close to  $\bar{A}$ , and  $\overline{A+B}$  can be factored into sparse lower and upper matrices  $\bar{L}$  and  $\bar{U}$ . The matrix  $\bar{B}$  could then be used to solve the iterative equation

$$(\overline{A+B})h^m = \bar{q} + \bar{B}h^{m-1}. \quad (96)$$

To reduce rounding errors, equation 96 can be reformulated. Subtracting the term  $(\overline{A+B})h^{m-1}$  from both sides yields

$$(\overline{A+B})(h^m - h^{m-1}) = \bar{q} - \bar{A}h^{m-1}. \quad (97)$$

Replacing the matrix  $\overline{A+B}$  with the product  $\bar{L}\bar{U}$  yields

$$\bar{L}\bar{U}(h^m - h^{m-1}) = \bar{q} - \bar{A}h^{m-1}. \quad (98)$$

The vector  $\overline{RES^m}$  can be defined by the equation

$$\overline{RES^m} = \bar{q} - \bar{A}h^{m-1}. \quad (99)$$

Then equation 96 can be written

$$\bar{L}\bar{U}(h^m - h^{m-1}) = \overline{RES^m}. \quad (100)$$

The problem of solving the equation

$$\bar{A}\bar{h} = \bar{q} \quad (101)$$

for the head distribution  $\bar{h}$  has been converted to iteratively solving equation 100. Matrices  $\bar{L}$  and  $\bar{U}$  can be derived from equations 95(a-m). Vector  $\bar{RES}^m$  can be calculated from the vectors  $\bar{q}$ ,  $\bar{h}^{m-1}$  (head calculated at the previous iteration) and the coefficient matrix  $\bar{A}$ . Equation 100 can be solved using first "forward" and then "back" substitution.

The first step is to perform forward substitution on the matrix equation

$$\bar{L}\bar{v} = \bar{RES}^m \quad (102)$$

solving for  $\bar{v}$ , where  $\bar{v} = \bar{U}(\bar{h}^m - \bar{h}^{m-1})$ . Then perform back substitution on the equation

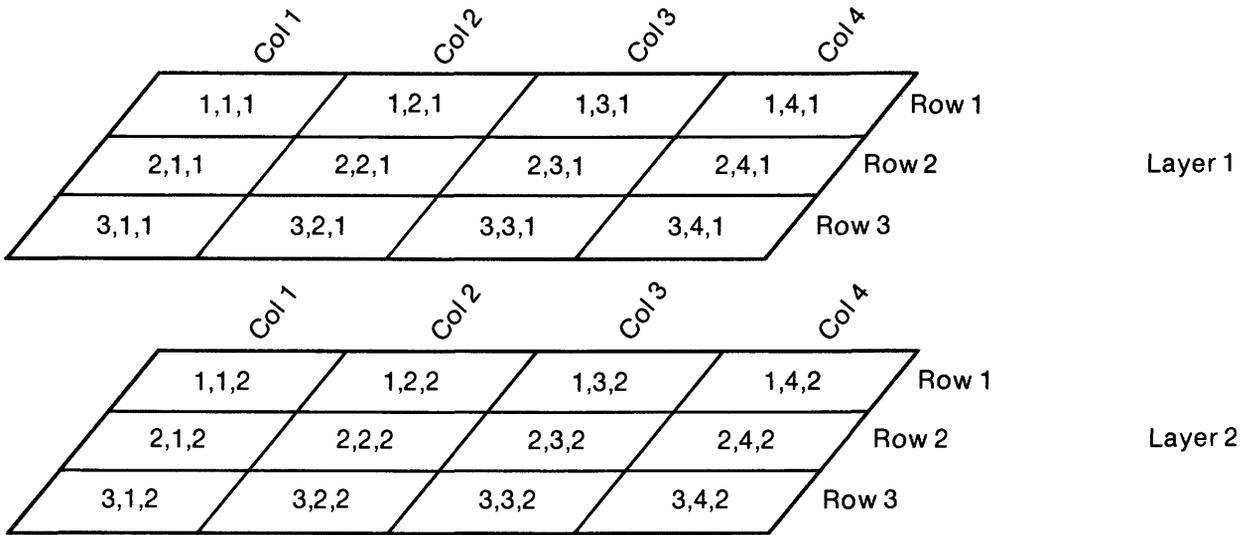
$$\bar{U}(\bar{h}^m - \bar{h}^{m-1}) = \bar{v} \quad (103)$$

solving for  $(\bar{h}^m - \bar{h}^{m-1})$ . In earlier discussions, the coefficients of the equations and hence the elements of the matrices are identified by the indices of the cells. At this point, it is convenient to renumber the equations sequentially starting with one (fig. 53).

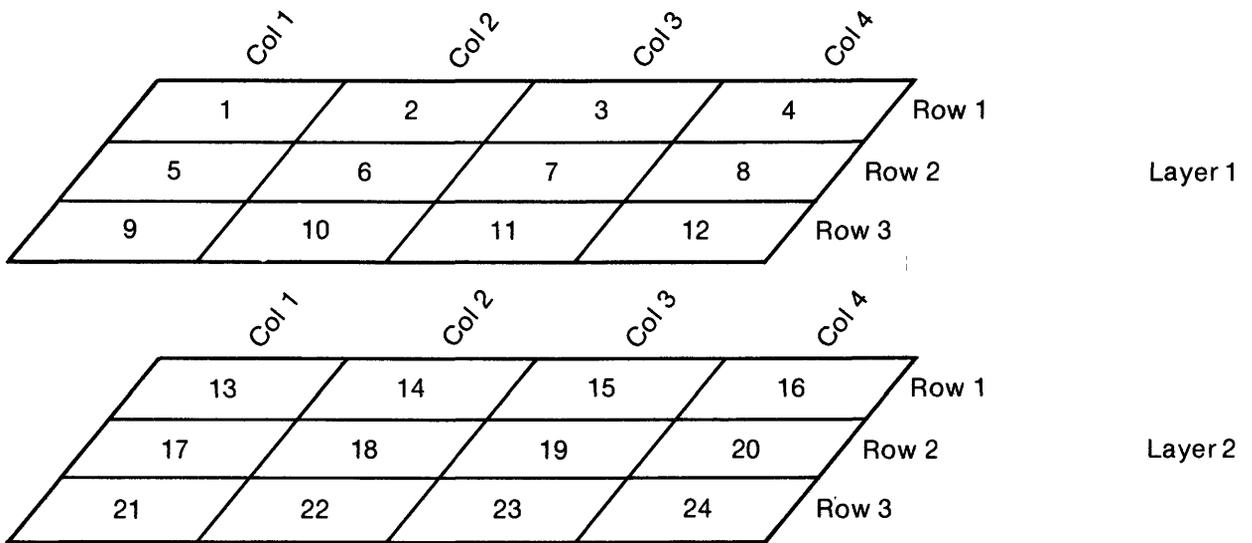
Because all elements in  $\bar{L}$  to the right of the main diagonal (fig. 50(a)) are equal to zero, the first linear equation represented by matrix equation 102 is

$$d_1 v_1 = RES_1^m. \quad (104)$$

Cell Numbering With 3 Indices



Cell Numbering With 3 Indices



Cell Numbering With 1 Index

Figure 53.—Cell numbering schemes for a grid using three indices and using one index.

The only unknown is  $v_1$ . Dividing both sides of equation 104 by  $d_1$  gives the value of  $v_1$ . The second linear equation represented by matrix equation 102 is

$$c_2v_1 + d_2v_2 = RES_2^m . \quad (105)$$

The value of  $v_1$  was calculated using the first equation; the only unknown is  $v_2$ --it too can be found by simple algebra. Similarly, all of the other elements of the vector  $\bar{v}$  can be calculated using the general equation

$$v_n = (\overline{RES}_n^m - a_nv_n - NRC - b_nv_n - NCOL - c_nv_{n-1})/d_n . \quad (106)$$

The reason equation 93 can be solved so easily is because the matrix  $\bar{L}$  has nothing but zeros to the right of the main diagonal--it is lower triangular. Thus starting at the top, each linear equation contains only one value in  $\bar{v}$  that was not calculated in an earlier equation. The technique used here to solve equation 102 for  $\bar{v}$  is called forward substitution.

Back substitution is similar to forward substitution. Since  $\bar{U}$  is upper triangular, however, the order of solution is reversed. The result of back substitution applied to equation 103 is the vector  $(\bar{h}^m - \bar{h}^{m-1})$ . That vector can be added to the vector  $\bar{h}^{m-1}$  to get the vector  $\bar{h}^m$ . The vector  $\bar{h}^m$  is the  $m$ -th estimate of the head distribution  $\bar{h}$  which is, in turn, a solution to the equation

$$\bar{A}\bar{h} = \bar{q} . \quad (107)$$

In summary, SIP is an iterative procedure which calculates a sequence of head distributions  $\bar{h}^1, \bar{h}^2, \dots, \bar{h}^m$  which converge to  $\bar{h}$ , the solution of  $\bar{A}\bar{h} = \bar{q}$ . Each head distribution consists of a value of head for each active cell. A head distribution  $\bar{h}^m$  is obtained by calculating a lower triangular

matrix  $\bar{L}$  and an upper triangular matrix  $\bar{U}$  such that  $\bar{LU}$  is "close" to  $\bar{A}$ .

Then the equation

$$\bar{LU}(h^m - h^{m-1}) = q - Ah^{m-1} \quad (108)$$

is solved for the head difference,  $\bar{h}^m - \bar{h}^{m-1}$ , using first forward and then back substitution. The head difference is added to  $\bar{h}^{m-1}$  to get  $\bar{h}^m$ .

Two further modifications of the procedure must be described. There are times when the method consistently overestimates the vector  $\bar{h}^m - \bar{h}^{m-1}$ . That overestimation may prevent convergence to  $\bar{h}$ . An acceleration parameter (ACCL) between zero and one has been introduced which multiplies the right side of equation 108. It provides the user with a means of dampening the overestimation.

Experience has shown that if the finite-difference equations are solved in two different orders on alternate iterations, the number of iterations needed to converge to a solution is lower than it would be if just one order were used. The order used in the discussion, so far, for solving the equations, has been to start at the first column, the first row, and the first layer and to proceed in ascending column order, ascending row order, and ascending layer order. An alternative is to start at the first column, the last row, and the last layer and to proceed in ascending column order, descending row order, and descending layer order. Using the same ordering of diagonal names used in figure 51, equations similar to equations 97(a-m) can be developed. They are

$$a_{i,j,k} = Z_{i,j,k} / (1 + \alpha e_{i,j,k+1} + \beta f_{i,j,k+1}) \quad (109a)$$

$$b_{i,j,k} = B_{i,j,k} / (1 + \gamma e_{i+1,j,k} + \beta g_{i+1,j,k}) \quad (109b)$$

$$c_{i,j,k} = D_{i,j,k} / (1 + \gamma f_{i,j-1,k} + \alpha g_{i,j-1,k}) \quad (109c)$$

$$A'_{i,j,k} = a_{i,j,k} e_{i,j,k+1} \quad (109d)$$

$$C'_{i,j,k} = e_{i+1,j,k} b_{i,j,k} \quad (109e)$$

$$G'_{i,j,k} = f_{i,j-1,k} c_{i,j,k} \quad (109f)$$

$$R'_{i,j,k} = g_{i,j-1,k} c_{i,j,k} \quad (109g)$$

$$T'_{i,j,k} = a_{i,j,k} f_{i,j,k+1} \quad (109h)$$

$$U'_{i,j,k} = b_{i,j,k} g_{i+1,j,k} \quad (109i)$$

$$\begin{aligned} d_{i,j,k} = & E_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k} \\ & + \gamma C'_{i,j,k} + \gamma G'_{i,j,k} + \beta U'_{i,j,k} \\ & + \alpha R'_{i,j,k} - a_{i,j,k} g_{i,j,k+1} - b_{i,j,k} f_{i+1,j,k} \\ & - e_{i,j-1,k} c_{i,j,k} \end{aligned} \quad (109j)$$

$$e_{i,j,k} = (F_{i,j,k} - \alpha A'_{i,j,k} - \gamma C'_{i,j,k}) / d_{i,j,k} \quad (109k)$$

$$f_{i,j,k} = (H_{i,j,k} - \beta T'_{i,j,k} - \gamma G'_{i,j,k}) / d_{i,j,k} \quad (109l)$$

$$g_{i,j,k} = (S_{i,j,k} - \alpha R'_{i,j,k} - \beta U'_{i,j,k}) / d_{i,j,k} \quad (109m)$$

Implementation of the concepts presented in the previous section entails setting up and solving either equations 95(a-m) or 109(a-m) to get the elements of matrices  $\bar{L}$  and  $\bar{U}$ , solving for the vectors  $\bar{v}$  and  $(\bar{h}^m - \bar{h}^{m-1})$ , and finally calculating the vector  $\bar{h}^m$ .

The first step is to select the three sequences of iteration parameters  $\alpha$ ,  $\beta$ , and  $\gamma$ . Selection of those parameters is based primarily on experience. Selection of the iteration parameters does not alter the value of the solution, only the efficiency of the solution method. It has been observed that all three sequences can be set equal to each other. Thus we will use a single sequence designated  $\omega$ .

Although iteration parameters are an important part of the SIP solution process, how to select the parameters is not well understood. The number of parameters, their values, and the order of cycling through them must be chosen. Weinstein, Stone, and Kwan (1969) use the same set of parameters for all three planes and use from 4 to 10 parameters. Numerous methods of cycling the parameters have been used. Trescott, Pinder, and Larson (1976) tested different parameter ordering in two-dimensional problems and found that repeated cycling through the parameters in the order of smallest to largest worked well for a variety of problems. That method of cycling is used in this program.

The selection of the iteration parameters themselves is the most critical for the efficient operation of SIP. It has been customary to base the selection of parameters on one minus the maximum parameter, called the "seed" in this report. The equation for the iteration parameters is

$$\omega(\ell) = 1 - \text{SEED}^{\ell-1/L-1} \quad \ell = 1, 2, \dots, \text{NPARM} \quad (110)$$

where

$\ell$  is an index going from 1 to the number of iteration parameters;

$\omega(\ell)$  is the  $\ell$ th iteration parameter;

NPARM is the number of iteration parameters; and

L is the number of iteration parameters.

Using this equation leaves only the seed to be specified. This choice is not straightforward and, accordingly, the user is given the choice of specifying the seed or permitting the model to calculate it.

The equation used to calculate the seed in the model is based on the equation developed to define the minimum iteration parameter for the Alternating Direction Implicit Procedure (ADIP) solution process. That the ADIP equation has any value for use with SIP is simply an empirical result (Weinstein, Stone, and Kwan, 1969). The equation is

$$\text{SEED} = \underset{\text{of}}{\text{minimum}} \left[ 2\text{NCOL} \frac{\pi^2}{2(1+\rho_1)}, 2\text{NROW} \frac{\pi^2}{2(1+\rho_2)}, 2\text{NLAY} \frac{\pi^2}{2(1+\rho_3)} \right], \quad (111)$$

$$\text{where } \rho_1 = \frac{\text{KC DELR}^2}{\text{KR DELC}^2} + \frac{\text{KV DELR}^2}{\text{KR DELV}^2} \quad (112)$$

$$\rho_2 = \frac{\text{KR DELC}^2}{\text{KC DELR}^2} + \frac{\text{KV DELC}^2}{\text{KC DELV}^2} \quad (113)$$

$$\rho_3 = \frac{\text{KR DELV}^2}{\text{KV DELR}^2} + \frac{\text{KC DELV}^2}{\text{KV DELC}^2} \quad (114)$$

where

NCOL is the number of columns in the model grid;

NROW is the number of rows in the model grid;

NLAY is the number of layers in the model grid;

KC is the hydraulic conductivity in the column direction;

KR is the hydraulic conductivity in the row direction;

KV is the hydraulic conductivity in the vertical direction;

DELC is the cell width in the column direction;

DELR is the cell width in the row direction; and

DELV is the cell width in the vertical direction.

The ADIP equation was developed assuming a uniform grid and homogeneous medium; that is, KC, KR, KV, DELR, DELC, and DELV were assumed constant. For non-uniform grids or nonhomogeneous mediums, Weinstein, Stone, and Kwan (1969) calculate equation 33 at every cell and take the overall minimum. In this

program, an average of the values calculated by applying equation 111 to every cell is used. Using the average prevents a single cell with unusual properties from dominating the calculation.

The  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  terms are modified to use conductance producing

$$\rho_1 = \frac{CC + CV}{CR} \quad (115)$$

$$\rho_2 = \frac{CR + CV}{CC} \quad (116)$$

$$\rho_3 = \frac{CR + CC}{CV} \quad (117)$$

where

CC is the cell conductance in the column direction;

CR is the cell conductance in the row direction; and

CV is the cell conductance in the vertical direction.

Equations 115-117 require conductances at a cell; however, these values are not known to the model program. Only conductances between nodes are defined. Therefore, equations 115-117 are rewritten in terms of conductances between nodes. Wherever a value of cell conductance is required in equations 115-117, one of the two values of conductance between the two neighboring nodes in the required direction is selected so that the resulting seed is minimized.

The resulting equations defining  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  are

$$\rho_1 = \frac{\text{MAX}(B,H) + \text{MAX}(Z,S)}{\text{MIN}(D,F)} \quad (118)$$

$$\rho_2 = \frac{\text{MAX}(D,F) + \text{MAX}(Z,S)}{\text{MIN}(B,H)} \quad (119)$$

$$\rho_3 = \frac{\text{MAX}(D,F) + \text{MAX}(B,H)}{\text{MIN}(Z,S)} \quad (120)$$

where

MAX is a mathematical function specifying the maximum of the values in parentheses;

MIN is a similar function specifying the minimum of values; and

B, H, Z, S, D, and F are the conductances as used in equation 80.

Experience has shown that setting the acceleration parameter (ACCL) to one and using the calculated value of the seed often results in inefficient convergence. Optimum convergence occurs when the convergence criterion is met with the least number of iterations. Convergence deviates from the optimum if the absolute value of head change each iteration is consistently either too small or too large. When head change is consistently too large, head overshoots the correct value and oscillations of head occurs. The sign of the head change repeatedly reverses in order to compensate for the overshoot. Severe overshoot causes divergence while moderate overshoot simply slows down convergence. When head change is consistently too small, the opposite problem occurs; head approaches the correct value slowly. In severe situations, the head-change criterion for convergence may be met, but the head will be far from the correct value. In such situations, a significant volumetric budget imbalance will occur.

Weinstein, Stone, and Kwan (1969) suggest that a trial and error method be used to improve the choice of the seed. Thus provisions have been made to permit the user to specify the seed. By observing the rate of convergence for several different seeds, an optimal seed can be selected.

The similarity between equations 95(a-m) and 109(a-m) was used to produce a system of equations with single indices taking on different values to

reflect the ordering of the equations. Since element  $v_n$  of the  $v$  vector can be calculated as soon as the  $n$ -th row of the  $\bar{L}$  matrix and  $\bar{U}$  matrix has been calculated, we can include the equation for  $\bar{v}_n$  as the last equation in the list. The index  $n11$  refers to the cell in the last layer to be calculated which is in the same row and column as cell  $n$ . The indices  $nr1$  and  $nc1$  are defined analogously. The equations are

$$a_n = Z_n / (1 + \omega(e_{n11} + f_{n11})) \quad (121a)$$

$$b_n = B_n / (1 + \omega(e_{nr1} + g_{nr1})) \quad (121b)$$

$$c_n = D_n / (1 + \omega(f_{nc1} + g_{nc1})) \quad (121c)$$

$$A'_n = a_n e_{n11} \quad (121d)$$

$$C'_n = b_n e_{nr1} \quad (121e)$$

$$G'_n = c_n f_{nc1} \quad (121f)$$

$$R'_n = c_n g_{nc1} \quad (121g)$$

$$T'_n = a_n f_{n11} \quad (121h)$$

$$U'_n = b_n g_{nr1} \quad (121i)$$

$$d_n = E_n + \omega(A'_n + T'_n + C'_n + G'_n + U'_n + R'_n) - a_n g_{n11} - b_n f_{nr1} - c_n e_{nc1} \quad (121j)$$

$$e_n = (F_n - \omega(A'_n + C'_n)) / d_n \quad (121k)$$

$$f_n = (H_n - \omega(T'_n + G'_n)) / d_n \quad (121l)$$

$$g_n = (S_n - \omega(R'_n + U'_n)) / d_n \quad (121m)$$

$$v_n = (RES_n - a_n v_{n11} - b_n v_{nr1} - c_n v_{nc1}) / d_n \quad (121n)$$

Since the backward substitution requires all values of  $e_n$ ,  $f_n$ ,  $g_n$ , and  $v_n$ , space will have to be allocated in the SIP Package for four arrays to store those values. Each array has as many elements as there are cells in the grid.

The coefficient matrix  $\bar{A}$  is sparse with only seven nonzero diagonals. Rather than passing an entire matrix to the SIP Package, only the nonzero diagonals are passed. Because of symmetry of the matrix, only the main diagonal and the three lower diagonals are needed. The three lower diagonals correspond to the conductance arrays CC, CR, and CV. The main diagonal is formed from the three conductance arrays and the array HCOF described in chapter 2. The right hand side of the matrix equation,  $\bar{q}$ , corresponds to the array RHS described in chapter 2. The latest estimate of the head distribution,  $\bar{h}^{m-1}$ , corresponds to the array HNEW. As new estimates of head are calculated by SIP, they are stored in HNEW replacing the previous estimates. Thus input to SIP consists of the following arrays: CC, CR, CV, RHS, HCOF, and HNEW. Output from SIP consists of a new HNEW. As explained in chapter 3, the Formulate Procedure is inside the iteration loop; therefore, the input arrays may be modified at each iteration.

## Strongly Implicit Procedure Package Input

Input to the Strongly Implicit Procedure (SIP) Package is read from the unit specified in IUNIT(9).

FOR EACH SIMULATION

### SIP1AL

1. Data:    MXITER       NPARM  
Format:    I10           I10

### SIP1RP

2. Data:    ACCL           HCLOSE        IPCALC        WSEED        IPRSIP  
Format:    F10.0         F10.0        I10           F10.0        I10

### Explanation of Fields Used in Input Instructions

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

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

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

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

IPCALC--is a flag indicating where the iteration parameter seed will come from.

0 - the seed will be entered by the user.

1 - the seed will be calculated at the start of the simulation from problem parameters.

WSEED--is the seed for calculating iteration parameters. It is only specified if IPCALC is equal to zero.

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

SAMPLE INPUT TO THE SIP PACKAGE(USER SPECIFIES THE SEED)

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{MXITER, NPARM}	50 5
2	{ACCL, HCLOSE, IPCALC, WSEED, IPRSIP}	1. .01 0 .98 10

SAMPLE INPUT TO THE SIP PACKAGE(PROGRAM CALCULATES THE SEED)

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{MXITER, NPARM}	100 6
2	{ACCL, HCLOSE, IPCALC, WSEED, IPRSIP}	1. .01 1

## Module Documentation for the Strongly Implicit Procedure Package

The Strongly Implicit Procedure Package (SIP1) consists of three primary modules and two submodules. They are:

### Primary Modules

- SIP1AL      Allocates space for SIP work arrays.
- SIP1RP      Reads control information needed by the SIP  
                 Package and calculates iteration parameters if  
                 the seed is specified by the user.
- SIP1AP      Performs one iteration of the strongly implicit  
                 procedure.

### Submodules

- SSIP1P      Prints the largest head change for each iteration.
- SSIP1I      Calculates iteration parameters when the seed  
                 is calculated by the program.

### Narrative for Module SIPIAL

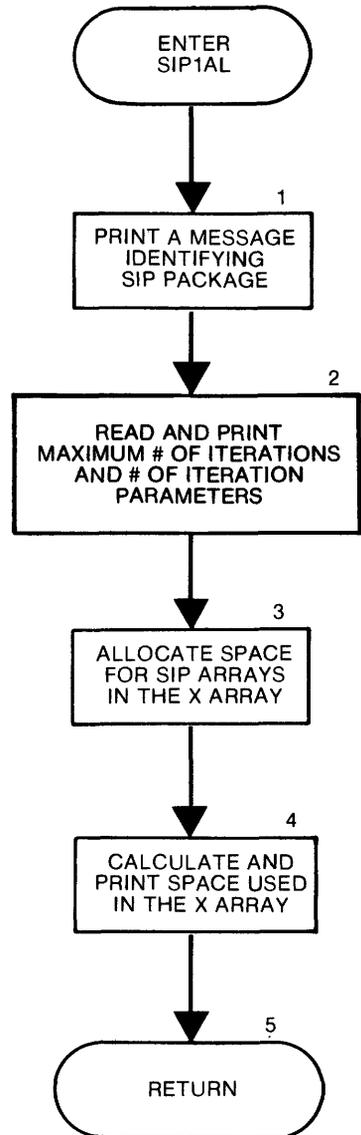
Module SIPIAL allocates space in the X array for SIP arrays. The four arrays, EL, FL, GL, and V hold intermediate results during the solution process. Each of these contains one element for each model cell. Additionally, three arrays, HDCG, LRCH, and W are required. HDCG holds the maximum head change each iteration and LRCH holds the cell location at which the maximum occurred. HDCG contains MXITER elements and LRCH contains three times MXITER elements, where MXITER--the maximum number of iterations allowed in a time step--is specified by the user. Array W holds iteration parameters. One element in W is used for each of the NPARM iteration parameters. (NPARM is specified by the user.)

Module SIPIAL performs its functions in the following order:

1. Print a message identifying the SIP Package.
2. Read and print MXITER and NPARM.
3. Allocate the required space in the X array. The X-array location pointer (ISUM) is saved in variable ISOLD prior to allocation so that the space required for SIP can be calculated in step 4.
4. Calculate and print the space used in the X array. The space used by SIP is  $ISUM - ISOLD$ . The total space allocated by all packages so far is  $ISUM - 1$ .
5. RETURN.

Flow Chart for Module SIP1AL

X array is the pool of memory space from which space is allocated for arrays used by various packages.



```

SUBROUTINE SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,LCHDCG,LCLRCH,
1 LCW,MXITER,NPARM,NCOL,NROW,NLAY,IN,IOUT)
C
C-----VERSION 1001 08DEC1983 SIP1AL
C
C *****
C ALLOCATE STORAGE IN THE X ARRAY FOR SIP ARRAYS
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----PRINT A MESSAGE IDENTIFYING SIP PACKAGE
WRITE(IOUT,1)IN
1 FORMAT(1H0,'SIP1 -- STRONGLY IMPLICIT PROCEDURE SOLUTION PACKAGE'
1,',' VERSION 1, 12/08/83',' INPUT READ FROM UNIT',I3)
C
C2-----READ AND PRINT MXITER AND NPARM
READ(IN,2) MXITER,NPARM
2 FORMAT(2I10)
WRITE(IOUT,3) MXITER,NPARM
3 FORMAT(1X,'MAXIMUM OF',I4,' ITERATIONS ALLOWED FOR CLOSURE'/
1 1X,I2,' ITERATION PARAMETERS')
C
C3-----ALLOCATE SPACE FOR THE SIP ARRAYS
ISOLD=ISUM
NRC=NROW*NCOL
ISIZ=NRC*NLAY
LCEL=ISUM
ISUM=ISUM+ISIZ
LCFL=ISUM
ISUM=ISUM+ISIZ
LCGL=ISUM
ISUM=ISUM+ISIZ
LCV=ISUM
ISUM=ISUM+ISIZ
LCHDCG=ISUM
ISUM=ISUM+MXITER
LCLRCH=ISUM
ISUM=ISUM+3*MXITER
LCW=ISUM
ISUM=ISUM+NPARM
C
C4-----CALCULATE AND PRINT THE SPACE USED IN THE X ARRAY
ISP=ISUM-ISOLD
WRITE(IOUT,4) ISP
4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED BY SIP')
ISUM1=ISUM-1
WRITE(IOUT,5) ISUM1,LENX
5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
IF(ISUM1.GT.LENX) WRITE(IOUT,6)
6 FORMAT(1X,' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C5-----RETURN
RETURN
END

```

List of Variables for Module SIPIAL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISIZ	Module	Number of cells in the grid.
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.
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	ISUM-1.
LCEL	Package	Location in the X array of the first element of array EL.
LCFL	Package	Location in the X array of the first element of array FL.
LCGL	Package	Location in the X array of the first element of array GL.
LCHDCG	Package	Location in the X array of the first element of array HDCG.
LCLRCH	Package	Location in the X array of the first element of array LRCH.
LCV	Package	Location in the X array of the first element of array V.
LCW	Package	Location in the X array of the first element of array W.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXITER	Package	Maximum number of iterations.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NPARM	Package	Number of iteration parameters.
NRC	Module	Number of cells in a layer.
NROW	Global	Number of rows in the grid.

## Narrative for Module SIP1RP

Module SIP1RP reads data for the SIP package: the acceleration parameter (ACCL), the closure criterion (HCLOSE), the iteration-parameter seed (WSEED), a flag indicating whether WSEED is to be calculated or specified by the user (IPCALC), and the interval for printing head change (IPRSIP). If IPCALC is zero, iteration parameters are calculated using WSEED as the seed. Module SIP1RP performs its functions in the following order:

1. Read the data. If ACCL is zero, substitute a default of 1.0. If IPRSIP is less than or equal to zero, substitute an interval of 999 time steps. The defaults are provided as a convenience to the user.

2. Print the data read in step 1.

3. Check IPCALC which is a flag that indicates the source of the iteration-parameter seed (WSEED).

- (a) If IPCALC is not zero, submodule SSIPII will calculate a seed and the resulting iteration parameters at the start of the first iteration. Print a message telling of this option.

- (b) If IPCALC is zero, use WSEED to calculate iteration parameters. The  $i$ -th iteration parameter ( $I_i$ ) is given by the expression

$$I_i = 1 - (\text{WSEED})^{\frac{i-1}{\text{NPARM}-1}}.$$

Print the parameters.

4. RETURN.

## Flow Chart for Module SIP1RP

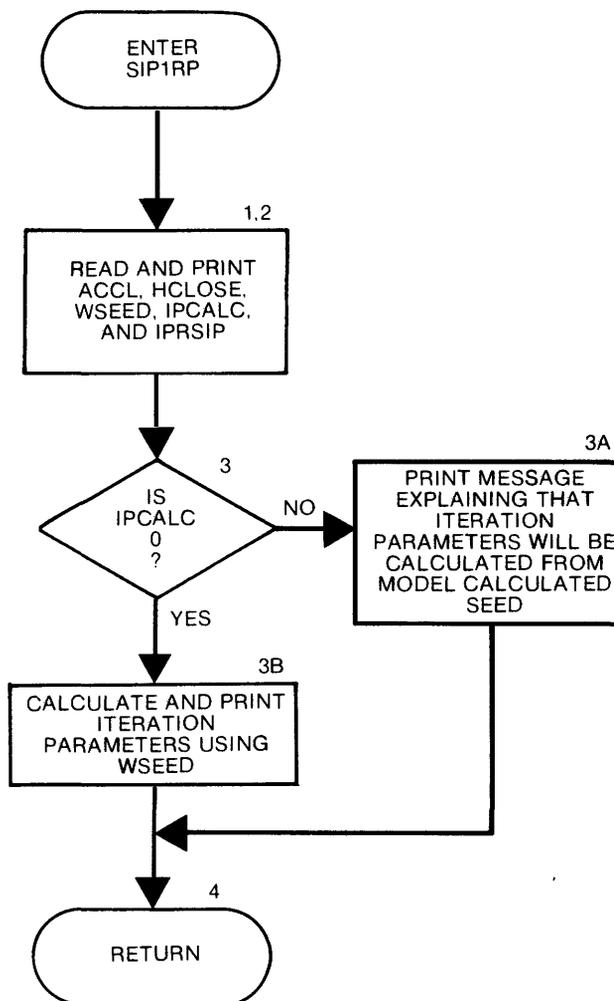
ACCL is a multiplier of calculated head change which is used to control the convergence rate.

HCLOSE is the head change closure criterion. When head change in all model cells is less than or equal to HCLOSE, iteration stops.

WSEED is the seed, specified by the user, on which the calculation of iteration parameters is based if IPCALC is zero.

IPRSIP is the time step interval for printing the maximum head change for each iteration of a time step. Head change is printed every IPRSIP time step. Head change is printed at the end of a stress period regardless of the interval.

IPCALC is a flag. If it is set equal to one, iteration parameters will be calculated from a seed calculated within the program. If it is clear (equal to zero), iteration parameters will be calculated from a seed provided by the user.



```

      SUBROUTINE SIPIRP(NPARM,MXITER,ACCL,HCLOSE,W,IN,IPCALC,IPRSIP,
1      IOUT)
C
C-----VERSION 0925 16DEC1982 SIPIRP
C
C *****
C READ DATA FOR SIP
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION W(NPARM)
C -----
C
C1-----READ ACCL,HCLOSE,WSEED,IPCALC,IPRSIP
      READ(IN,1) ACCL,HCLOSE,IPCALC,WSEED,IPRSIP
1      FORMAT(2F10.0,I10,F10.0,I10)
      IF(ACCL.EQ.0.) ACCL=1.
C
C2-----PRINT DATA VALUES JUST READ
      WRITE(IOUT,100)
100  FORMAT(1H0,///57X,'SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE'
1/57X,43('-'))
      WRITE(IOUT,115) MXITER
115  FORMAT(1H0,47X,'MAXIMUM ITERATIONS ALLOWED FOR CLOSURE =',I9)
      WRITE(IOUT,120) ACCL
120  FORMAT(1H ,63X,'ACCELERATION PARAMETER =',G15.5)
      WRITE(IOUT,125) HCLOSE
125  FORMAT(1H ,52X,'HEAD CHANGE CRITERION FOR CLOSURE =',E15.5)
      IF(IPRSIP.LE.0)IPRSIP=999
      WRITE(IOUT,130) IPRSIP
130  FORMAT(1H ,52X,'SIP HEAD CHANGE PRINTOUT INTERVAL =',I9)
C
C3-----CHECK IF SPECIFIED VALUE OF WSEED SHOULD BE USED OR IF
C3-----SEED SHOULD BE CALCULATED
      IF(IPCALC.EQ.0) GO TO 150
C
C3A-----CALCULATE SEED & ITERATION PARAMETERS PRIOR TO 1ST ITERATION
      WRITE(IOUT,140)
140  FORMAT(1H0,52X,'CALCULATE ITERATION PARAMETERS FROM MODEL',
1' CALCULATED WSEED')
      GO TO 1000
C
C3B-----USE SPECIFIED VALUE OF WSEED
C3B-----CALCULATE AND PRINT ITERATION PARAMETERS
150  P1=-1.
      P2=NPARM-1
      DO 160 I=1,NPARM
      P1=P1+1.
160  W(I)=1.-WSEED**(P1/P2)
      WRITE(IOUT,161) NPARM,WSEED,(W(J),J=1,NPARM)
161  FORMAT(1H0,/,I5,' ITERATION PARAMETERS CALCULATED FROM',
1' SPECIFIED WSEED =',F11.8,' :',(10X,6E15.7))
C
C4-----RETURN
1000 RETURN
      END

```

List of Variables for Module SIP1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ACCL	Package	Acceleration parameter.
HCLOSE	Package	Closure criterion for the iterative procedure.
I	Module	Index for iteration parameters.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPCALC	Package	Flag.  ≠ 0, seed for iteration will be calculated in the program. = 0, seed will be specified by the user.
IPRSIP	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
J	Module	Index for iteration parameters.
MXITER	Package	Maximum number of iterations.
NPARAM	Package	Number of iteration parameters.
P1	Module	I - 1.
P2	Module	NPARAM - 1.
W	Package	DIMENSION (NPARAM), Iteration parameters.
WSEED	Module	Seed for calculating iteration parameters.

## Narrative for Module SIPIAP

Module SIPIAP performs one iteration of the Strongly Implicit Procedure (SIP) algorithm for solving the flow equation. To save computational time, all arrays are declared one dimensional. The one-dimensional indexes are calculated from the layer, row, and column indexes normally used to access the arrays in three dimensions. Computational time is saved because knowledge of the geometry is used to make the calculations efficient and because the calculations are not repeated for identical indexes as would be done by internal FORTRAN addressing routines if three-dimensional subscripts were used.

This module is complex, partly because the SIP solution process requires that the same calculations be performed with two methods of ordering the equations. This is implemented by a generalized algorithm that uses the same computer statements to handle both ordering schemes. Checks are made to detect which ordering scheme is used, and array indexes are calculated accordingly.

Double precision is used for most calculations in this module in order to allow accurate answers to be calculated for a wide range of problems. Mixed precision arithmetic has been avoided by setting double-precision variables equal to single-precision values and then using the double-precision variables to make a completely double-precision expression.

Also, the reverse has been done. In the explanations below, each use of an assignment statement to change precision is not explained because of the large amount of text required. It is left to the reader to discern when this is occurring. When changing this module, care should be used

to maintain expressions that have unmixed precision. Mixed precision expressions can cause erroneous results with some compilers.

Module SIP1AP performs its functions in the following order:

1. If the user has specified (IPCALC  $\neq$  0) that iteration parameters should be calculated by the program, CALL submodule SSIPII to calculate both the seed and the parameters.
2. Assign values to fields that are constant during an iteration.
3. Initialize the variables that track maximum head change during an iteration.
4. Clear SIP work arrays.
5. Determine the ordering of equations and set the ordering flag (IDIR) accordingly. This flag alternates between 1 and -1 each iteration. Calculate indexes IDNRC and IDNCOL which are used when calculating locations of neighboring cells.
6. Calculate the matrix  $\bar{U}$  and intermediate vector  $\bar{V}$  using forward substitution. The elements in matrix  $\bar{L}$  are used as they are calculated; therefore, they are not saved. In the explanation of SIP concepts, the diagonals in the matrix  $\bar{U}$  were designated "e," "f," and "g." The corresponding field names in the program are EL (e lower case), FL (f lower case), and GL (g lower case). Similarly, the diagonals in the  $\bar{L}$  array which are "a," "b," "c," and "d" in the explanation are "AL," "BL," "CL," and "DL" in the program. The codes for the diagonals in matrix  $\bar{A}$  in the explanation are the same in the program. The codes for diagonals in  $\overline{A+B}$  in the explanation are followed by a "P" in the program. Hence, Z' in the

explanation is ZP (Z prime) in the program. The intermediate vector  $\bar{V}$  in the explanation is the array "V" in the program.

(a) Set current cell indexes, II, JJ, KK. For normal ordering, the equation order is the same as the order of the loop indexes I,J,K. For reverse ordering, loop indexes I and K are inverted to produce the proper sequence of cells.

(b) Calculate the one-dimensional subscript of the current cell. If this cell is constant head or no flow, skip calculations for this cell and go on to the next.

(c) Calculate the one-dimensional subscripts for the six neighboring cells.

(d) Calculate the one-dimensional subscripts for conductance to each of the six neighboring cells. Since conductances between cells are assigned to array elements at specific cells (for example, CR(I,J,K) stores conductance between cells I,J,K and I,J+1,K), the four or five conductance subscripts are not simply the cell locations of the six neighboring cells as calculated in step 6(c). Also, the subscripts depend on equation ordering.

(e) Calculate or assign variables that are required for forward substitution and involve neighboring cells. Whenever a neighboring cell is outside of the grid, the variables are set to zero.

- (1) Neighboring cell is one row back.
- (2) Neighboring cell is one row ahead.
- (3) Neighboring cell is one column back.

(4) Neighboring cell is one column ahead.

(5) Neighboring cell is one layer back.

(6) Neighboring cell is one layer ahead.

(f) Calculate the components of the upper and lower matrices  $\bar{U}$  and  $\bar{L}$ , which are the factors of matrix  $\bar{A+B}$ .

(g) Calculate the residual  $\bar{RES}$ . The calculation of HNW times HCOF is done in single precision so that the calculation will have precision comparable to similar calculations made in the formulation modules, all of which use single precision.

(h) Calculate the intermediate vector  $\bar{V}$ , which is stored in array V. This step completes the forward-substitution process for one cell.

7. Step through the cells solving for head change using back substitution.

(a) Set current cell indexes II, JJ, KK. The ordering is the reverse of that used for forward substitution (step 6(a)).

(b) Calculate the one-dimensional subscript of the current cell. If this cell is constant head or no flow, skip calculations for this cell and go to the next.

(c) Calculate the one-dimensional subscripts for the three neighboring cells behind (relative to the direction of the back-substitution ordering) the current cell.

(d) Back substitute, solving for head change. Store head change in array V in place of the intermediate values of vector  $\bar{V}$ . This doubling up of storage is used to save the cost of additional computer storage.

(e) Save the value of head change whose absolute value is largest during this iteration. Also, save the cell location where this head change occurred and the absolute value of the head change.

(f) Add the head change this iteration to head from the previous iteration to get a new estimate of head.

8. Store the head change whose absolute value is greatest this iteration and its cell location in arrays HDCG and LRCH. These may be printed in step 10 at the end of the time step. Set the convergence flag to one if the convergence criterion is met.

9. If the iteration is complete, print the number of iterations for the step; otherwise, RETURN.

10. Print the maximum head change and cell location each iteration if the SIP printout interval (IPRSIP) is reached. Printout occurs at the end of a stress period regardless of the interval.

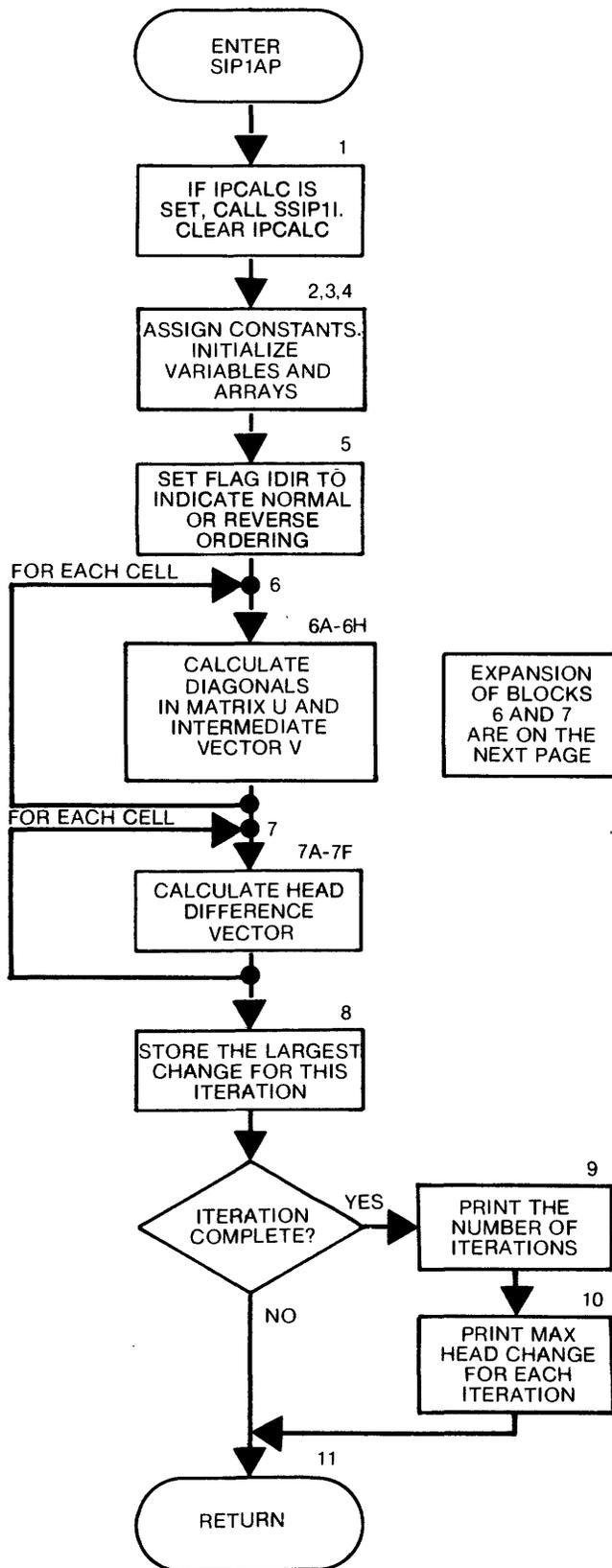
11. RETURN.

## Flow Chart for Module SIP1AP

IPCALC is a flag. If it is set equal to one, the program calculates a seed from which iteration parameters are calculated. It may be set by the user at the beginning of the simulation. It is cleared during the first iterations. SSIPII will never be called more than once. If IPCALC is not set equal to zero, the user specifies the seed for the iteration parameters.

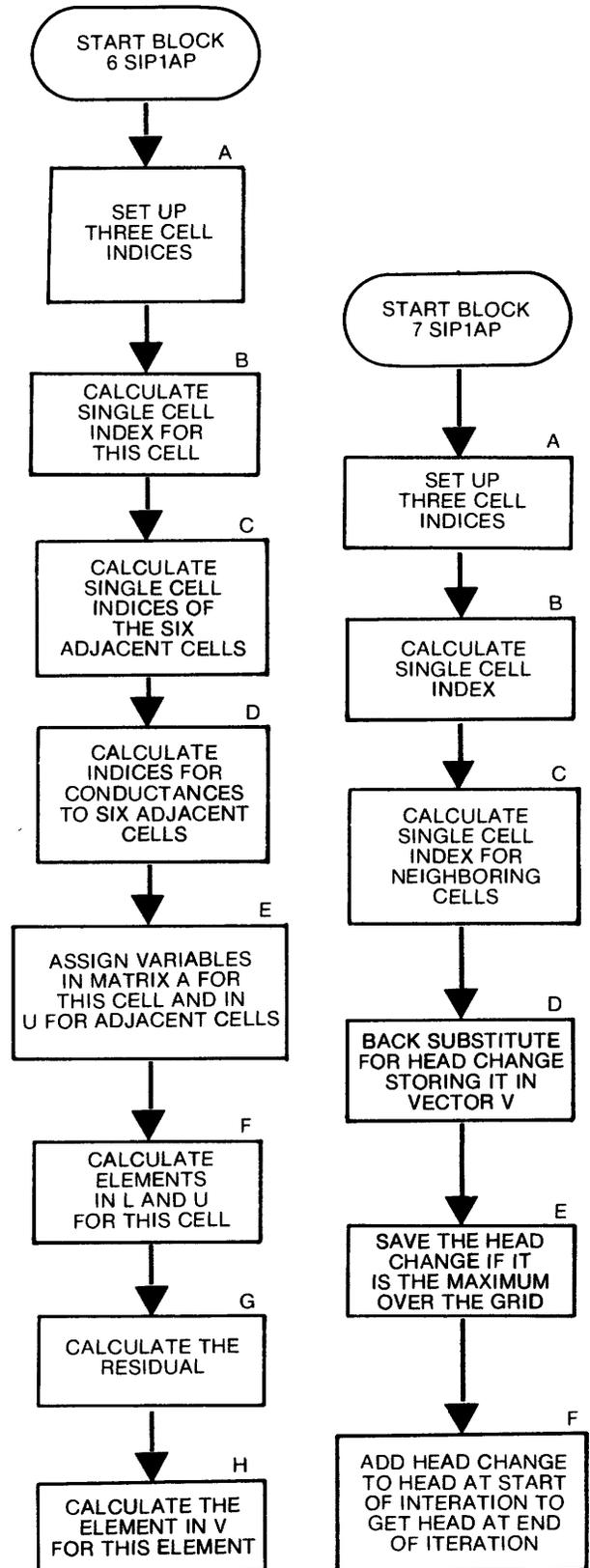
IDIR indicates whether the ordering of equations is normal (1) or reverse (-1).

SSIPII is a submodule which calculates iteration parameters.



Flow Chart for Module SIP1AP (Continued)

Single Cell Index: In this module, a single index is used to identify each cell. This is in opposition to the three indices (I,J,K) used in most other modules.



```

SUBROUTINE SIPIAP(HNEW,IBOUND,CR,CC,CV,HCOF,RHS,EL,FL,GL,V,
1      W,HDCG,LRCH,NPARM,KITER,HCLOSE,ACCL,ICNVG,KSTP,KPER,
2      IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,IOUT)
C-----VERSION 1454 02DEC1982 SIPIAP
C
C *****
C SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE -- 1 ITERATION
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW,DITPAR,AC,HCOF,RRHS,XI,DZERO,DONE,RES
C DOUBLE PRECISION Z,B,D,E,F,H,S,AP,TP,CP,GP,UP,RP
C DOUBLE PRECISION ZHNEW,BHNEW,DHNEW,FHNEW,HHNEW,SHNEW
C DOUBLE PRECISION AL,BL,CL,DL,ELNCL,FLNCL,GLNCL
C DOUBLE PRECISION ELNRL,FLNRL,GLNRL,ELNLL,FLNLL,GLNLL
C DOUBLE PRECISION VNRL,VNCL,VNLLI,ELXI,FLXI,GLXI,VN,HCFHNW
C
C DIMENSION HNEW(NODES),IBOUND(NODES),CR(NODES),CC(NODES),
1 CV(NODES),HCOF(NODES),RHS(NODES),EL(NODES),FL(NODES),
2 GL(NODES),V(NODES),W(NPARM),HDCG(MXITER),LRCH(3,MXITER)
C -----
C
C1-----CALCULATE ITERATION PARAMETERS IF FLAG IS SET. THEN
C1-----CLEAR THE FLAG SO THAT CALCULATION IS DONE ONLY ONCE.
      IF(IPCALC.NE.0)
1        CALL SSIPI1(CR,CC,CV,IBOUND,NPARM,W,NCOL,NROW,NLAY,IOUT)
      IPCALC=0
C
C2-----ASSIGN VALUES TO FIELDS THAT ARE CONSTANT DURING AN ITERATION
      DZERO=0.
      DONE=1.
      AC=ACCL
      NRC=NROW*NCOL
      NTH=MOD(KITER-1,NPARM)+1
      DITPAR=W(NTH)
C
C3-----INITIALIZE VARIABLE THAT TRACKS MAXIMUM HEAD CHANGE DURING
C3-----THE ITERATION
      BIGG=0.
C
C4-----CLEAR SIP WORK ARRAYS.
      DO 100 I=1,NODES
        EL(I)=0.
        FL(I)=0.
        GL(I)=0.
      100 V(I)=0.
C
C5-----SET NORMAL/REVERSE EQUATION ORDERING FLAG (1 OR -1) AND
C5-----CALCULATE INDEXES DEPENDENT ON ORDERING
      IDIR=1
      IF(MOD(KITER,2).EQ.0)IDIR=-1
      IDNRC=IDIR*NRC
      IDNCOL=IDIR*NCOL
C
C6-----STEP THROUGH CELLS CALCULATING INTERMEDIATE VECTOR V
C6-----USING FORWARD SUBSTITUTION
      DO 150 K=1,NLAY
        DO 150 I=1,NROW
          DO 150 J=1,NCOL
C
C6A-----SET UP CURRENT CELL LOCATION INDEXES. THESE ARE DEPENDENT
C6A-----ON THE DIRECTION OF EQUATION ORDERING.
      IF(IDIR.LE.0)GO TO 120
      II=I
      JJ=J
      KK=K
      GO TO 122
      120 II=NROW-I+1
      JJ=J

```

```

        KK=NLAY-K+1
C
C6B-----CALCULATE 1 DIMENSIONAL SUBSCRIPT OF CURRENT CELL AND
C6B-----SKIP CALCULATIONS IF CELL IS NOFLOW OR CONSTANT HEAD
        122 N=JJ+(II-1)*NCOL+(KK-1)*NRC
           IF(IBOUND(N),LE.0)GO TO 150
C
C6C-----CALCULATE 1 DIMENSIONAL SUBSCRIPTS FOR LOCATING THE 6
C6C-----SURROUNDING CELLS
        NRN=N+IDNCOL
        NRL=N-IDNCOL
        NCN=N+1
        NCL=N-1
        NLN=N+IDNRC
        NLL=N-IDNRC
C
C6D-----CALCULATE 1 DIMENSIONAL SUBSCRIPTS FOR CONDUCTANCE TO THE 6
C6D-----SURROUNDING CELLS. THESE DEPEND ON ORDERING OF EQUATIONS.
        IF(IDIR.LE.0)GO TO 124
        NCF=N
        NCD=NCL
        NRB=NRL
        NRH=N
        NLS=N
        NLZ=NLL
        GO TO 126
    124 NCF=N
        NCD=NCL
        NRB=N
        NRH=NRN
        NLS=NLN
        NLZ=N
C
C6E-----ASSIGN VARIABLES IN MATRICES A & U INVOLVING ADJACENT CELLS
C6E1-----NEIGHBOR IS 1 ROW BACK
    126 B=DZERO
        ELNRL=DZERO
        FLNRL=DZERO
        GLNRL=DZERO
        BHNEW=DZERO
        VNRL=DZERO
        IF(I.EQ.1) GO TO 128
        B=CC(NRB)
        ELNRL=EL(NRL)
        FLNRL=FL(NRL)
        GLNRL=GL(NRL)
        BHNEW=B*HNEW(NRL)
        VNRL=V(NRL)
C
C6E2-----NEIGHBOR IS 1 ROW AHEAD
    128 H=DZERO
        HHNEW=DZERO
        IF(I.EQ.NROW) GO TO 130
        H=CC(NRH)
        HHNEW=H*HNEW(NRN)
C
C6E3-----NEIGHBOR IS 1 COLUMN BACK
    130 D=DZERO
        ELNCL=DZERO
        FLNCL=DZERO
        GLNCL=DZERO
        DHNEW=DZERO
        VNCL=DZERO
        IF(J.EQ.1) GO TO 132
        D=CR(NCD)
        ELNCL=EL(NCL)
        FLNCL=FL(NCL)
        GLNCL=GL(NCL)
        DHNEW=D*HNEW(NCL)
        VNCL=V(NCL)
C
C6E4-----NEIGHBOR IS 1 COLUMN AHEAD

```

```

132 F=DZERO
    FHNEW=DZERO
    IF(J.EQ.NCOL) GO TO 134
    F=CR(NCF)
    FHNEW=F*HNEW(NCN)
C
C6E5----NEIGHBOR IS 1 LAYER BEHIND
134 Z=DZERO
    ELNLL=DZERO
    FLNLL=DZERO
    GLNLL=DZERO
    ZHNEW=DZERO
    VNLL=DZERO
    IF(K.EQ.1) GO TO 136
    Z=CV(NLZ)
    ELNLL=EL(NLL)
    FLNLL=FL(NLL)
    GLNLL=GL(NLL)
    ZHNEW=Z*HNEW(NLL)
    VNLL=V(NLL)
C
C6E6----NEIGHBOR IS 1 LAYER AHEAD
136 S=DZERO
    SHNEW=DZERO
    IF(K.EQ.NLAY) GO TO 138
    S=CV(NLS)
    SHNEW=S*HNEW(NLN)
C
C6E7----CALCULATE THE NEGATIVE SUM OF ALL CONDUCTANCES TO NEIGHBORING
C6E7----CELLS
138 E=-Z-B-D-F-H-S
C
C6F----CALCULATE COMPONENTS OF THE UPPER AND LOWER MATRICES, WHICH
C6F----ARE THE FACTORS OF MATRIX (A+B)
    AL=Z/(DONE+DITPAR*(ELNLL+FLNLL))
    BL=B/(DONE+DITPAR*(ELNRL+GLNRL))
    CL=D/(DONE+DITPAR*(FLNCL+GLNCL))
    AP=AL*ELNLL
    CP=BL*ELNRL
    GP=CL*FLNCL
    RP=CL*GLNCL
    TP=AL*FLNLL
    UP=BL*GLNRL
    HCOF=HCOF(N)
    DL=E+HCOF+DITPAR*(AP+TP+CP+GP+UP+RP)-AL*GLNLL-BL*FLNRL-CL*ELNCL
    EL(N)=(F-DITPAR*(AP+CP))/DL
    FL(N)=(H-DITPAR*(TP+GP))/DL
    GL(N)=(S-DITPAR*(RP+UP))/DL
C
C6G----CALCULATE THE RESIDUAL
    RRHS=RHS(N)
    HNW=HNEW(N)
    HCFHNW=HNW*HCOF(N)
    RES=RRHS-ZHNEW-BHNEW-DHNEW-E*HNEW(N)-HCFHNW-FHNEW-HHNEW-SHNEW
C
C6H----CALCULATE THE INTERMEDIATE VECTOR V
    V(N)=(AC*RES-AL*VNLL-BL*VNRL-CL*VNCL)/DL
C
150 CONTINUE
C
C7-----STEP THROUGH EACH CELL AND SOLVE FOR HEAD CHANGE BY BACK
C7-----SUBSTITUTION
    DO 160 K=1,NLAY
    DO 160 I=1,NROW
    DO 160 J=1,NCOL
C
C7A----SET UP CURRENT CELL LOCATION INDEXES. THESE ARE DEPENDENT
C7A----ON THE DIRECTION OF EQUATION ORDERING.
    IF(IDIR.LT.0) GO TO 152
    KK=NLAY-K+1
    II=NROW-I+1
    JJ=NCOL-J+1

```

```

        GO TO 154
152 KK=K
        II=I
        JJ=NCOL-J+1
C
C7B-----CALCULATE 1 DIMENSIONAL SUBSCRIPT OF CURRENT CELL AND
C7B-----SKIP CALCULATIONS IF CELL IS NOFLOW OR CONSTANT HEAD
        154 N=JJ+(II-1)*NCOL+(KK-1)*NRC
        IF(IBOUND(N).LE.0)GO TO 160
C
C7C-----CALCULATE 1 DIMENSIONAL SUBSCRIPTS FOR THE 3 NEIGHBORING CELLS
C7C-----BEHIND (RELATIVE TO THE DIRECTION OF THE BACK SUBSTITUTION
C7C-----ORDERING) THE CURRRENT CELL.
        NC=N+1
        NR=N+IDNCOL
        NL=N+IDNRC
C
C7D-----BACK SUBSTITUTE, STORING HEAD CHANGE IN ARRAY V IN PLACE OF
C7D-----INTERMEDIATE FORWARD SUBSTITUTION VALUES.
        ELXI=DZERO
        FLXI=DZERO
        GLXI=DZERO
        IF(JJ.NE.NCOL) ELXI=EL(N)*V(NC)
        IF(I.NE.1) FLXI=FL(N)*V(NR)
        IF(K.NE.1) GLXI=GL(N)*V(NL)
        VN=V(N)
        V(N)=VN-ELXI-FLXI-GLXI
C
C7E-----GET THE ABSOLUTE HEAD CHANGE. IF IT IS MAX OVER GRID SO FAR.
C7E-----THEN SAVE IT ALONG WITH CELL INDICES AND HEAD CHANGE.
        TCHK=ABS(V(N))
        IF (TCHK.LE.BIGG) GO TO 155
        BIGG=TCHK
        BIG=V(N)
        IB=II
        JB=JJ
        KB=KK
C
C7F-----ADD HEAD CHANGE THIS ITERATION TO HEAD FROM THE PREVIOUS
C7F-----ITERATION TO GET A NEW ESTIMATE OF HEAD.
        155 XI=V(N)
        HNEW(N)=HNEW(N)+XI
C
        160 CONTINUE
C
C8-----STORE THE LARGEST ABSOLUTE HEAD CHANGE (THIS ITERATION) AND
C8-----AND ITS LOCATION.
        HDCG(KITER)=BIG
        LRCH(1,KITER)=KB
        LRCH(2,KITER)=IB
        LRCH(3,KITER)=JB
        ICNVG=0
        IF(BIGG.LE.HCLOSE) ICNVG=1
C
C9-----IF END OF TIME STEP, PRINT # OF ITERATIONS THIS STEP
        IF(ICNVG.EQ.0 .AND. KITER.NE.MXITER) GO TO 600
        IF(KSTP.EQ.1) WRITE(IOUT,500)
        500 FORMAT(1H0)
        WRITE(IOUT,501) KITER,KSTP,KPER
        501 FORMAT(1X,I5,' ITERATIONS FOR TIME STEP',I4,' IN STRESS PERIOD',
        1          I3)
C
C10-----PRINT HEAD CHANGE EACH ITERATION IF PRINTOUT INTERVAL IS REACHED
        IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP .OR. MOD(KSTP,IPRSIP).EQ.0)
        1          CALL SSIPIP(HDCG,LRCH,KITER,KSTP,KPER,MXITER,IOUT)
C
C11-----RETURN
600 RETURN
C
        END

```

List of Variables for Module SIPIAP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
AC	Module	Double-precision acceleration parameter (ACCL).
ACCL	Package	Acceleration parameter.
AL	Module	Diagonal from the lower factor. (AL stands for A-lower case.)
AP	Module	Diagonal element in the modified coefficient matrix. (AP stands for A-prime.)
B	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the last row.
BHNEW	Module	Head in the adjacent cell which is in the last row.
BIG	Module	Largest head change for an iteration.
BIGG	Module	Largest absolute value of head change for an iteration.
BL	Module	Diagonal from the lower factor. (BL stands for B-lower case.)
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CL	Module	Diagonal from the lower factor. (CL stands for C-lower case.)
CP	Module	Diagonal element in the modified coefficient matrix. (CP stands for C-prime.)
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K)
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
D	Module	Diagonal label in the coefficient matrix. Conductance from the adjacent node which is in the last column.
DHNEW	Module	Head in the adjacent cell which is in the last column.
DITPAR	Module	Double-precision iteration parameter.
DL	Module	Diagonal from the lower factor. (DL stands for D-lower case.)
DONE	Module	Double-precision field containing a one.
DZERO	Module	Double-precision field containing a zero.
E	Module	Main diagonal in the coefficient matrix.
EL	Module	DIMENSION (NODES), Diagonal from the upper factor. (EL stands for E-lower case.)
ELNCL	Module	EL (E-lower case) from the cell in the last column.
ELNLL	Module	EL (E-lower case) from the cell in the last layer.
ELNRL	Module	EL (E-lower case) from the cell in the last row.
ELXI	Module	Intermediate result.
F	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the next column.
FHNEW	Module	Head in the adjacent cell which is in the next column.
FL	Module	DIMENSION (NODES), Diagonal from the upper factor. (FL stands for F-lower case.)
FLNCL	Module	FL (F-lower case) from the cell in the last column.
FLNLL	Module	FL (F-lower case) from the cell in the last layer.
FLNRL	Module	FL (F-lower case) from the cell in the last row.

List of Variables for Module SIPIAP (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
FLXI	Module	Intermediate result.
GL	Module	DIMENSION (NODES), Diagonal from the upper factor. (GL stands for G-lower case.)
GLNCL	Module	GL (G-lower case) from the cell in the last column.
GLNLL	Module	GL (G-lower case) from the cell in the last layer.
GLNRL	Module	GL (G-lower case) from the cell in the last row.
GLXI	Module	Intermediate result.
GP	Module	Diagonal element in the modified coefficient matrix. (GP stands for G-prime.)
H	Module	Diagonal label in the coefficient matrix. Conductance from the adjacent node which is in the next row.
HCFHNW	Module	Product of head and HCOF for a cell.
HCLOSE	Package	Closure criterion for the iterative procedure.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HDCG	Package	DIMENSION (MXITER), Maximum head change for each iteration.
HHCOF	Module	Double-precision HCOF.
HHNEW	Module	Head in the adjacent cell which is in the next row.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HNW	Module	Temporary field for HNEW(N).
I	Module	Index for nodes and rows.
IB	Module	Row number of the cell having the largest head change.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
ICNVG	Global	Flag is set equal to one when the iteration procedure has converged.
IDIR	Module	Indicator for direction of solution algorithm. +1 - forward -1 - reverse
IDNCOL	Module	Intermediate result used to calculate indices.
IDNRC	Module	Intermediate result used to calculate indices.
II	Module	Row number.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPCALC	Package	Flag. = 0, iteration parameter seed (WSEED) is entered by the user. = 1, seed is calculated in the program.
IPRSIP	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
J	Module	Index for columns.
JB	Module	Column number of the cell having the largest head change.
JJ	Module	Column index.
K	Module	Index for layers.
KB	Module	Layer of the cell having the largest head change.
KITER	Global	Iteration counter. Reset at the start of each time step.

List of Variables for Module SIPIAP (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KK	Module	Layer index.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
LRCH	Package	DIMENSION (MXITER), Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MXITER	Package	Maximum number of iterations.
N	Module	Cell index.
NC	Module	Index for the adjacent cell in the last column.
NCD	Module	One-dimensional subscript of conductance to the adjacent cell which is in the last column.
NCF	Module	One-dimensional subscript of conductance to the adjacent cell which is in the next column.
NCL	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the last column.
NCN	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the next column.
NCOL	Global	Number of columns in the grid.
NL	Module	Index for the adjacent cell in the last layer.
NLAY	Global	Number of layers in the grid.
NLL	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the last layer.
NLN	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the next layer.
NLS	Module	One-dimensional subscript of conductance to the adjacent cell which is in the next layer.
NLZ	Module	One-dimensional subscript of conductance to the adjacent cell which is in the last layer.
NODES	Global	Number of cells (nodes) in the finite-difference grid.
NPARAM	Package	Number of iteration parameters.
NR	Module	Index for the adjacent cell in the last row.
NRB	Module	One-dimensional subscript of conductance to the adjacent cell which is in the last row.
NRC	Module	Number of cells in the layer.
NRH	Module	One-dimensional subscript of conductance to the adjacent cell which is in the next row.
NRL	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the last row.
NRN	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the next row.
NROW	Global	Number of rows in the grid.
NSTP	Global	Number of time steps in the current stress period.
NTH	Module	Index for iteration parameters.
RES	Module	Residual.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

List of Variables for Module SIPIAP (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
RP	Module	Diagonal element in the modified coefficient matrix. (RP stands for R-prime.)
RRHS	Module	Double-precision right hand side of the equation.
S	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the next layer.
SHNEW	Module	Head in the adjacent cell which is in the next layer.
TCHK	Module	Absolute value of head change for a single cell.
TP	Module	Diagonal element in the modified coefficient matrix. (TP stands for T-prime.)
UP	Module	Diagonal element in the modified coefficient matrix. (UP stands for U-prime.)
V	Package	DIMENSION (NODES), Intermediate result.
VN	Module	Temporary double-precision V(N).
VNCL	Module	Element in the intermediate vector for the cell in the last column.
VNLL	Module	Element in the intermediate vector for the cell in the last layer.
VNLLI	Module	Artifact of an earlier version of the program.
VNRL	Module	Element in the intermediate vector for the cell in the last row.
W	Package	DIMENSION (NPARM), Iteration parameters.
XI	Module	Double-precision V(N).
Z	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the last layer.
ZHNEW	Module	Head in the adjacent cell which is in the last layer.

### Narrative for Module SSIP1P

Submodule SSIP1P prints the largest value of head change (HDCG) out of all cells for each iteration of a time step. Also printed is the cell location (LRCH) where the change occurs. The submodule is so short that no numbered comments are used and no flow chart is provided.

```

SUBROUTINE SSI1P(HDCG,LRCH,KITER,KSTP,KPER,MXITER,IOUT)
C
C
C-----VERSION 1504 08DEC1982 SSI1P
C*****
C PRINT MAXIMUM HEAD CHANGE FOR EACH ITERATION DURING A TIME STEP
C*****
C
C SPECIFICATIONS:
C-----
C
C DIMENSION HDCG(MXITER), LRCH(3,MXITER)
C-----
C
C WRITE(IOUT,5)
C 5 FORMAT(1H0,'MAXIMUM HEAD CHANGE FOR EACH ITERATION: '/
C 1 1H0,5(' HEAD CHANGE LAYER,ROW,COL')/1X,132('-'))
C WRITE (IOUT,10) (HDCG(J),(LRCH(I,J),I=1,3),J=1,KITER)
C 10 FORMAT((1X,5(G12.4,' (' ,I3,' ',I3,' ',I3,' ')))
C WRITE(IOUT,11)
C 11 FORMAT(1H0)
C
C RETURN
C
C END

```

List of Variables for Module SSI1P

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
HDCG	Package	DIMENSION (MXITER), Maximum head change for each iteration.
I	Module	Index for cell location.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
J	Module	Index for iterations.
KITER	Global	Iteration counter. Reset at the start of each time step.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
LRCH	Package	DIMENSION (MXITER), Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MXITER	Package	Maximum number of iterations.

## Narrative for Module SSIP1I

Submodule SSIP1I calculates an iteration-parameter seed using model-conductance values and grid dimensions. Although a single seed is required, the method of calculation requires that three-directional seeds be calculated for each active cell. Then a cell seed, the minimum of the three, is selected. Finally, all the cell seeds are averaged to give the grid seed. This grid seed is then used to calculate the iteration parameters. The minimum cell seed is also printed.

Submodule SSIP1I performs its functions in the following order:

1. Calculate constants and initialize variables. In order to calculate the average cell seed, accumulators AVGSUM (sum of the cell seeds) and NODES (sum of the active cells for which a seed is calculated) are required. These are initialized to zero. WMINMN is used to store the smallest cell seed. Since this value must always be less than one, it is initialized to 1.0. The three coefficients, CCOL, CROW, and CLAY are set equal to  $\pi^2/2(\text{NCOL})^2$ ,  $\pi^2/2(\text{NROW})^2$ , and  $\pi^2/2(\text{NLAY})^2$ , respectively.

2. Loop through all cells, calculating a cell seed for each active cell.

- (a) Find the conductances from the cell to each of the six adjacent cells. Conductance across the grid boundary is set equal to zero.

- (b) Find the maximum and minimum of the two conductances in the row direction (DFMX, DFMN), in the column direction (BHMx, BHMN), and in the vertical direction (ZSMx, ZSMN). If the minimum is zero (which indicates that a neighbor is no flow), set the minimum equal to the maximum.

(c) Calculate three-directional seeds (WCOL, WROW, WLAY) using the relations

$$WCOL = CCOL / (1. + (BHMx + ZSMx) / DFMN);$$

$$WROW = CROW / (1. + (DFMX + ZSMx) / BHMN); \text{ and}$$

$$WLAY = CLAY / (1. + (DFMX + BHMx) / ZSMN).$$

If the minimum conductance is zero (that is, both the minimum and the maximum are zero), set the seed equal to 1.0. This value will be ignored when the cell seed (the minimum-directional seed) is selected in step 2(d) because any valid seed will be less than 1.0.

(d) Select the minimum of the three-directional seeds as the cell seed. If it is the smallest cell seed used so far, store it in WMINMN. Accumulate the sum of the cell seeds and the total number of active cells so that the average of all cell seeds can be calculated in step 3.

3. Calculate the grid seed (the average cell seed) and print it along with the minimum seed.

4. Calculate and print iteration parameters using the grid seed with the relation

$$I_i = 1 - (SEED)^{\frac{i-1}{NPARM - 1}}$$

where

$I_i$  is the  $i$ -th iteration parameter, and

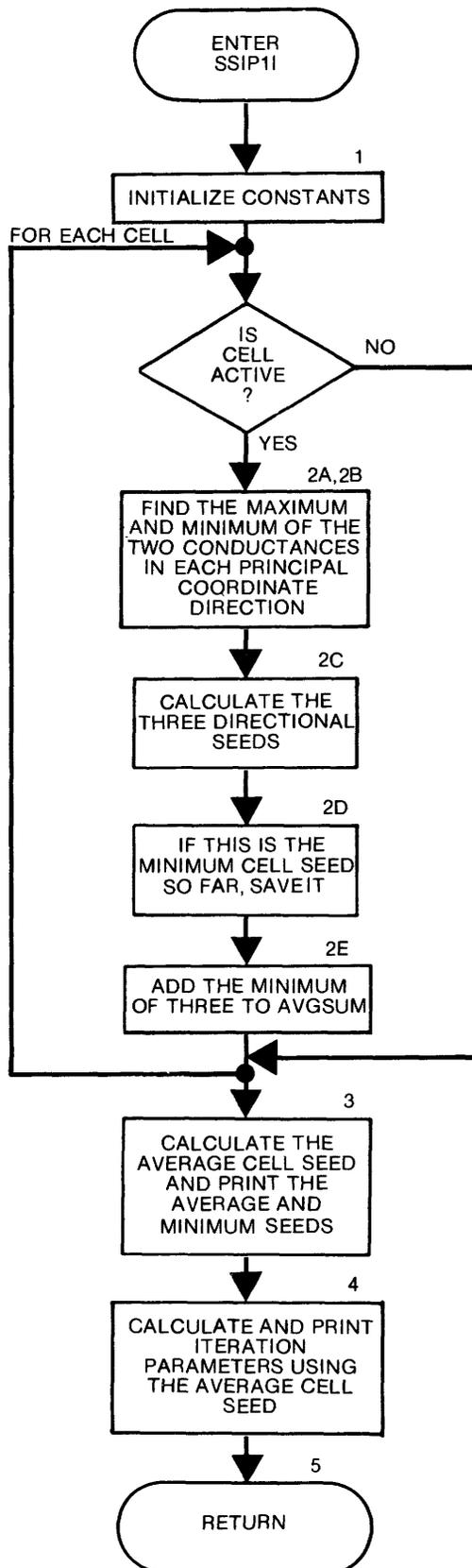
NPARM is the number of iteration parameters.

5. RETURN.

Flow Chart for Module SSIP11

Seed: the "grid seed" is the single parameter used to calculate the iteration parameters. To calculate the grid seed, several intermediate variables, called "cell seeds," are used. For each cell, three "directional seeds" are calculated. The minimum directional seed for a cell is the "cell seed." The "grid seed" is the average of the cell seeds.

AVGSUM is an accumulator to which each cell seed is added. It is then divided by the number of cells to obtain the average cell seed which is used as the grid seed.



```

SUBROUTINE SSIPI1(CR,CC,CV,IBOUND,NPARM,W,NCOL,NROW,NLAY,
1          IOUT)
C
C-----VERSION 1743 26APR1983 SSIPI1
C *****
C CALCULATE AN ITERATION PARAMETER SEED AND USE IT TO CALCULATE SIP
C ITERATION PARAMETERS
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION CR(NCOL,NROW,NLAY),CC(NCOL,NROW,NLAY)
1          ,CV(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),W(NPARM)
C
C DOUBLE PRECISION DWMIN,AVGSUM
C -----
C1-----CALCULATE CONSTANTS AND INITIALIZE VARIABLES
PIEPIE=9.869604
R=NROW
C=NCOL
L=NLAY
CCOL=PIEPIE/(2.*C*C)
CROW=PIEPIE/(2.*R*R)
CLAY=PIEPIE/(2.*L*L)
WMINMN=1.
AVGSUM=0.
NODES=0
C
C2-----LOOP THROUGH ALL CELLS, CALCULATING A SEED FOR EACH CELL
C2-----THAT IS ACTIVE
DO 100 K=1,NLAY
DO 100 I=1,NROW
DO 100 J=1,NCOL
IF(IBOUND(J,I,K).LE.0) GO TO 100
C
C2A-----CONDUCTANCE FROM THIS CELL
C2A-----TO EACH OF THE 6 ADJACENT CELLS
D=0.
IF(J.NE.1) D=CR(J-1,I,K)
F=0.
IF(J.NE.NCOL) F=CR(J,I,K)
B=0.
IF(I.NE.1) B=CC(J,I-1,K)
H=0.
IF(I.NE.NROW) H=CC(J,I,K)
Z=0.
IF(K.NE.1) Z=CV(J,I,K-1)
S=0.
IF(K.NE.NLAY) S=CV(J,I,K)
C
C2B-----FIND THE MAXIMUM AND MINIMUM OF THE 2 CONDUCTANCE COEFFICIENTS
C2B-----IN EACH PRINCIPAL COORDINATE DIRECTION
DFMX=AMAX1(D,F)

```

```

BHM=AMAX1(B,H)
ZSM=AMAX1(Z,S)
DFM=AMIN1(D,F)
BHMN=AMIN1(B,H)
ZSMN=AMIN1(Z,S)
IF(DFM.EQ.0.) DFM=DFM
IF(BHMN.EQ.0.) BHMN=BHM
IF(ZSMN.EQ.0.) ZSMN=ZSM

```

```

C
C2C-----CALCULATE A SEED IN EACH PRINCIPAL COORDINATE DIRECTION
      WCOL=1.
      IF(DFM.NE.0.) WCOL=CCOL/(1.+(BHM+ZSM)/DFM)
      WROW=1.
      IF(BHMN.NE.0.) WROW=CROW/(1.+(DFM+ZSM)/BHMN)
      WLAY=1.
      IF(ZSMN.NE.0.) WLAY=CLAY/(1.+(DFM+BHM)/ZSMN)

```

```

C
C2D-----SELECT THE CELL SEED, WHICH IS THE MINIMUM SEED OF THE 3.
C2D-----SELECT THE MINIMUM SEED OVER THE WHOLE GRID.
      WMIN=AMIN1(WCOL,WROW,WLAY)
      WMINMN=AMIN1(WMINMN,WMIN)

```

```

C
C2E-----ADD THE CELL SEED TO THE ACCUMULATOR AVGSUM FOR USE
C2E-----IN GETTING THE AVERAGE SEED.
      DWMIN=WMIN
      AVGSUM=AVGSUM+DWMIN
      NODES=NODES+1

```

```

C
100 CONTINUE

```

```

C
C3-----CALCULATE THE AVERAGE SEED OF THE CELL SEEDS, AND PRINT
C3-----THE AVERAGE AND MINIMUM SEEDS.
      TMP=NODES
      AVGMIN=AVGSUM
      AVGMIN=AVGMIN/TMP
      WRITE(IOUT,101) AVGMIN,WMINMN
101 FORMAT(1H0,'AVERAGE SEED =',F11.8/1X,'MINIMUM SEED =',F11.8)

```

```

C
C4-----CALCULATE AND PRINT ITERATION PARAMETERS FROM THE AVERAGE SEED
      P1=-1.
      P2=NPARM-1
      DO 50 I=1, NPARM
      P1=P1+1.
50 W(I)=1.-AVGMIN**(P1/P2)
      WRITE(IOUT,150) NPARM,(W(J),J=1, NPARM)
150 FORMAT(1H0,/,I5,' ITERATION PARAMETERS CALCULATED FROM',
1      ' AVERAGE SEED:'// (10X,6E15.7))

```

```

C
C5-----RETURN
      RETURN
      END

```

List of Variables for Module SSIP1I

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
AVGMIN	Module	Mean WMIN.
AVGSUM	Module	Sum of all of WMIN's.
B	Module	Conductance between this node and the one to the rear.
BHMN	Module	Minimum of B and H (if the minimum is 0, it is the maximum).
BHMX	Module	Maximum of B and H.
C	Module	Number of columns.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CCOL	Module	Intermediate factor.
CLAY	Module	Intermediate factor.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CROW	Module	Intermediate factor.
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
D	Module	Conductance between this node and the one to the left.
DFMN	Module	Minimum of D and F (if the minimum is 0, it is the maximum).
DFMX	Module	Maximum of D and F.
DWMIN	Module	Double precision WMIN.
F	Module	Conductance between this node and the one to the right.
H	Module	Conductance between this node and the one to the front.
I	Module	Index for rows.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell

List of Variables for Module SSIPII (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
J	Module	Index for columns.
K	Module	Index for layers.
L	Module	Number of layers.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NODES	Module	Number of variable-head (IBOUND > 0) cells in the grid.
NPARAM	Package	Number of iteration parameters.
NROW	Global	Number of rows in the grid.
PIEPIE	Module	PI squared.
P1	Module	Index for the number of parameters.
P2	Module	NPARAM-1.
R	Module	Number of rows.
S	Module	Conductance between this node and the one below.
TMP	Module	Temporary field for nodes.
W	Package	DIMENSION (NPARAM), Iteration parameters.
WCOL	Module	Seed in the column direction for a cell.
WLAY	Module	Seed in the layer direction for a cell.
WMIN	Module	Minimum of (WCOL, WLAY, WROW).
WMINMN	Module	Minimum WMIN.
WROW	Module	Seed in the row direction for a cell.
Z	Module	Conductance between this node and the one above.
ZSMN	Module	Minimum of Z and S (if minimum is 0, it is the maximum).
ZSMX	Module	Maximum of Z and S.

## CHAPTER 13

### SLICE-SUCCESSIVE OVERRELAXATION PACKAGE

#### Conceptualization and Implementation

Slice-successive overrelaxation (SSOR) is a technique for iteratively solving a system of linear equations. The SSOR package, in particular, solves the finite-difference equations associated with the cells in the finite-difference grid. Those equations with the iteration counter " $\ell$ " included have the form

$$\begin{aligned}
 & CV_{i,j,k-1/2} h_{i,j,k-1}^{m,\ell+1} + CC_{i-1/2,j,k} h_{i-1,j,k}^{m,\ell+1} + CR_{i,j-1/2,k} h_{i,j-1,k}^{m,\ell+1} \\
 & + (- CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
 & - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} - HCOF_{i,j,k}) h_{i,j,k}^{m,\ell+1} + CR_{i,j+1/2,k} h_{i,j+1,k}^{m,\ell+1} \\
 & + CC_{i+1/2,j,k} h_{i+1,j,k}^{m,\ell+1} + CV_{i,j,k+1/2} h_{i,j,k+1}^{m,\ell+1} = RHS_{i,j,k} \quad (122)
 \end{aligned}$$

where

$m$  is the time step counter; and

$\ell$  is the iteration counter.

Equation 122 can be written in residual form as

$$\begin{aligned}
 & CV_{i,j,k-1/2} (h_{i,j,k-1}^{m,\ell+1} - h_{i,j,k-1}^{m,\ell}) + CC_{i-1/2,j,k} (h_{i-1,j,k}^{m,\ell+1} - h_{i-1,j,k}^{m,\ell}) \\
 & + CR_{i,j-1/2,k} (h_{i,j-1,k}^{m,\ell+1} - h_{i,j-1,k}^{m,\ell}) + (- CV_{i,j,k-1/2} - CC_{i-1/2,j,k} \\
 & - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} \\
 & + HCOF_{i,j,k}) (h_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell}) + CR_{i,j+1/2,k} (h_{i,j+1,k}^{m,\ell+1} - h_{i,j+1,k}^{m,\ell}) \\
 & + CC_{i+1/2,j,k} (h_{i+1,j,k}^{m,\ell+1} - h_{i+1,j,k}^{m,\ell}) + CV_{i,j,k+1/2} (h_{i,j,k+1}^{m,\ell+1} - h_{i,j,k+1}^{m,\ell}) =
 \end{aligned}$$

$$\begin{aligned}
& \text{RHS}_{i,j,k} - \text{CV}_{i,j,k-1/2} h_{i,j,k-1}^{m,\ell} - \text{CC}_{i-1/2,j,k} h_{i-1,j,k}^{m,\ell} \\
& - \text{CR}_{i,j-1/2,k} h_{i,j-1,k}^{m,\ell} - (-\text{CV}_{i,j,k-1/2} - \text{CC}_{i-1/2,j,k} - \text{CR}_{i,j-1/2,k} \\
& - \text{CR}_{i,j+1/2,k} - \text{CC}_{i+1/2,j,k} - \text{CV}_{i,j,k+1/2} + \text{HCOF}_{i,j,k}) h_{i,j,k}^{m,\ell} \\
& - \text{CR}_{i,j+1/2,k} h_{i,j+1,k}^{m,\ell} - \text{CC}_{i+1/2,j,k} h_{i+1,j,k}^{m,\ell} - \text{CV}_{i,j,k+1/2} h_{i,j,k+1}^{m,\ell} \quad (123)
\end{aligned}$$

The left hand side of equation 123 consists of terms involving the seven head changes e.g.,  $(h_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell})$  for iteration  $\ell+1$ . At the beginning of iteration  $\ell+1$ , those head changes are unknown. The right hand side of equation 123 consists of the term RHS and terms involving the head at the end of iteration  $\ell$ , all of which are known at the beginning of iteration  $\ell+1$ . The difficulty encountered in solving the system of equations having the form of equation 123 is that there are too many equations--there is an equation for each cell in the grid. SSOR reduces the number of equations by simultaneously solving only those equations representing cells in a single slice (fig. 54). Note that the slice is identified by its row number; therefore, in this discussion, the terms row and slice will be used interchangeably.

Consider the equation for cells in row 1--that is, those equations for which  $i$  is equal to one. Since  $i-1$  will then be equal to zero, the heads  $h_{i-1,j,k}^{m,\ell+1}$  and  $h_{i-1,j,k}^{m,\ell}$  will be equal to zero. Therefore, the term  $\text{CC}_{i-1/2,j,k} (h_{i-1,j,k}^{m,\ell+1} - h_{i-1,j,k}^{m,\ell})$  is equal to zero. Furthermore, the head  $h_{i+1,j,k}^{m,\ell+1}$  which is the head in a cell in row 2, can be approximated--only while heads in row 1 are being calculated--by  $h_{i+1,j,k}^{m,\ell}$  (the head calculated at the previous iteration). Then the term  $\text{CC}_{i+1/2,j,k} (h_{i+1,j,k}^{m,\ell+1} - h_{i+1,j,k}^{m,\ell})$  is equal to zero. Thus equation 123 for cells in row 1 can be approximated by the equation

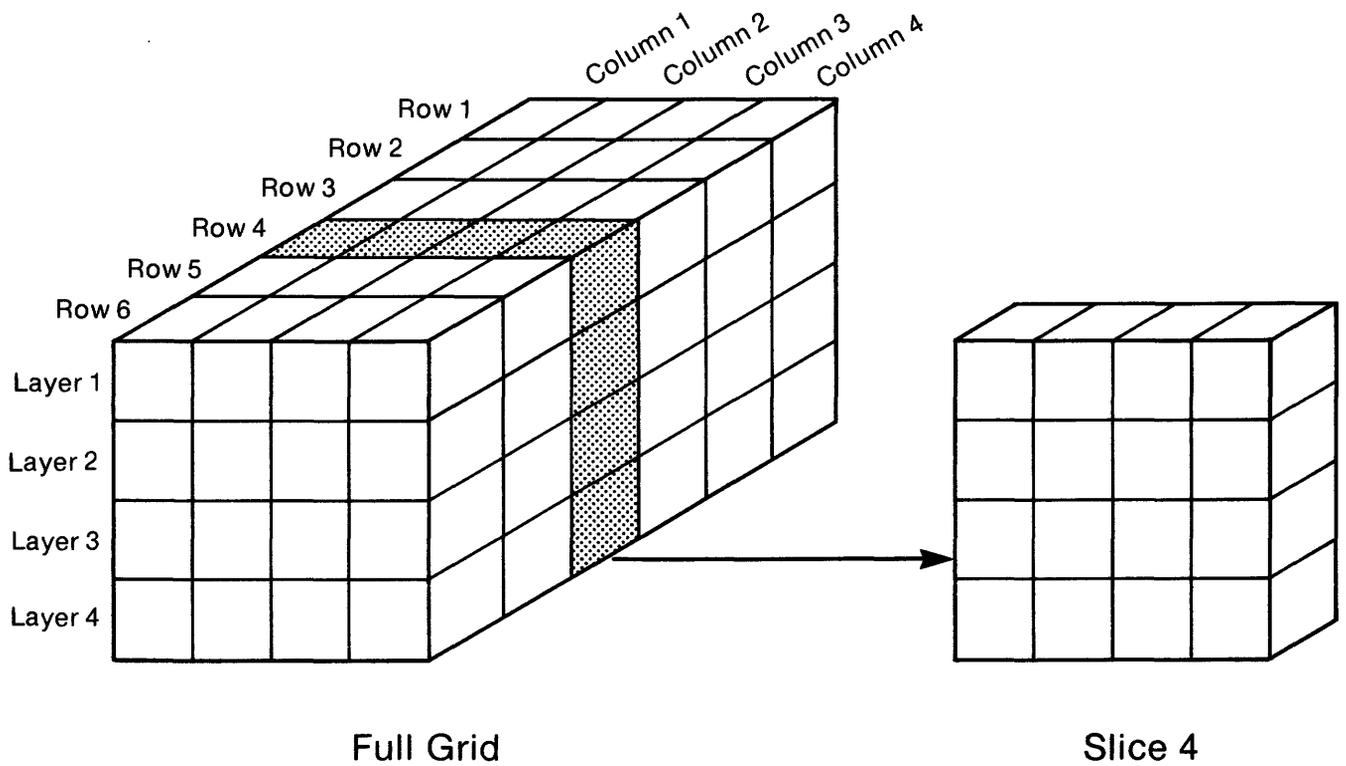


Figure 54.—SSOR reduces the number of equations that must be solved simultaneously by considering a single vertical slice at a time.

$$\begin{aligned}
& CV_{i,j,k-1/2,k} \tilde{h}_{i,j,k-1}^{m,\ell+1} - h_{i,j,k-1}^{m,\ell} + CR_{i,j-1/2,k} \tilde{h}_{i,j-1,k}^{m,\ell+1} - h_{i,j-1,k}^{m,\ell} \\
& + (- CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
& - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k}) \tilde{h}_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell} \\
& + CR_{i,j+1/2,k} \tilde{h}_{i,j+1,k}^{m,\ell+1} - h_{i,j+1,k}^{m,\ell} + CV_{i,j,k+1/2} \tilde{h}_{i,j,k+1}^{m,\ell+1} - h_{i,j,k+1}^{m,\ell} = \\
& RHS_{i,j,k} - CV_{i,j,k-1/2} h_{i,j,k-1}^{m,\ell} - CC_{i-1/2,j,k} h_{i-1,j,k}^{m,\ell+1} \\
& - CR_{i,j-1/2,k} h_{i,j-1,k}^{m,\ell} - (- CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} \\
& - CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k}) h_{i,j,k}^{m,\ell} \\
& - CR_{i,j+1/2,k} h_{i,j+1,k}^{m,\ell} - CC_{i+1/2,j,k} h_{i+1,j,k}^{m,\ell} - CV_{i,j,k+1/2} h_{i,j,k+1}^{m,\ell} \quad (124)
\end{aligned}$$

where  $\tilde{h}_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell}$  is approximately equal to  $h_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell}$ . The term  $\tilde{h}_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell}$  is a first estimate of head change for cell  $i,j,k$  at iteration  $\ell+1$ . The equations of the form of equation 123 for cells in row 1 can be replaced by equations of the form of equation 124. Those equations constitute a system of  $n$  simultaneous linear equations (where  $n$  is the number of cells in row 1) in  $n$  unknowns. In most ground-water problems,  $n$  is relatively small; therefore, the  $n$  equation can be solved directly using Gaussian elimination to get values of  $\tilde{h}_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell}$  in row 1. The value  $h_{i,j,k}^{m,\ell+1}$  can then be calculated by the equation

$$h_{i,j,k}^{m,\ell+1} = h_{i,j,k}^{m,\ell} + \omega(\tilde{h}_{i,j,k}^{m,\ell+1} - h_{i,j,k}^{m,\ell}) \quad (125)$$

where  $\omega$  is an acceleration parameter--generally between 1 and 2. Each of the rest of the slices can be handled in a similar manner. Terms in equations for slice  $P$  which involve heads in slice  $P+1$  use heads from

the preceding iteration. Thus the equations for slice P approximate equations of the form of equation 123. Solution of those equations gives approximate head changes for the current iteration. The approximate head changes are multiplied by the relaxation factor to get the final head changes for the iteration.

Implementation of slice-successive overrelaxation involves processing the rows one at a time, first reformulating the equations for each row from the form of equation 123 to the form of equation 124, and then solving the resulting equations to get the approximate head changes for each cell in the row. Finally, the head changes must be multiplied by the relaxation factor and added to the head at the previous iteration to get the new head value.

The equations for a row can be written in matrix form

$$\bar{A}\bar{X} = \bar{B} \tag{126}$$

where

$\bar{A}$  is the coefficient matrix;

$\bar{X}$  is the vector of head changes from iteration  $\ell$  to iteration  $\ell+1$ ; and

$\bar{B}$  is the vector of residuals.

The matrix  $\bar{A}$  is symmetric and banded (fig. 55) with a maximum half-bandwidth equal to the number of layers. Therefore, just the lower triangular portion of the coefficient matrix needs to be stored (fig. 55). It is stored in a rectangular array,  $\bar{A}$ , dimensioned by the maximum half-bandwidth plus one and the number of cells in a slice.

$a_{11}$	$a_{12}$		$a_{14}$					
$a_{12}$	$a_{22}$	$a_{23}$		$a_{25}$				
	$a_{23}$	$a_{33}$			$a_{36}$			
$a_{14}$			$a_{44}$	$a_{45}$		$a_{47}$		
	$a_{25}$		$a_{45}$	$a_{55}$	$a_{56}$			
		$a_{36}$		$a_{56}$	$a_{66}$		$a_{68}$	
			$a_{47}$			$a_{77}$		
					$a_{68}$		$a_{88}$	$a_{89}$
							$a_{89}$	$a_{99}$

Full Matrix

$a_{11}$	$a_{22}$	$a_{33}$	$a_{44}$	$a_{55}$	$a_{66}$	$a_{77}$	$a_{88}$	$a_{99}$
$a_{12}$	$a_{23}$		$a_{45}$	$a_{56}$			$a_{89}$	
					$a_{68}$			
$a_{14}$	$a_{25}$	$a_{36}$	$a_{47}$					

Compressed Matrix

Figure 55.—Organization of elements in a matrix for a slice in conventional notation and in compressed notation.

At the end of each iteration, the maximum head change for the iteration is compared to the closure criterion. If the maximum head change is smaller than the closure criterion, a flag (ICNVG) is set equal to one. The flag is returned to the main program. If it is set, the main program terminates the iteration loop.

## Slice-Successive Overrelaxation Package Input

Input to the Slice-Successive Overrelaxation (SOR) Package is read from the unit specified in IUNIT(11).

FOR EACH SIMULATION

SOR1AL

1. Data: MXITER  
Format: I10

SOR1RP

2. Data: ACCL HCLOSE IPRSOR  
Format: F10.0 F10.0 I10

### Explanation of Fields Used in Input Instructions

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

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

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

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



## Narrative for Module SORIAL

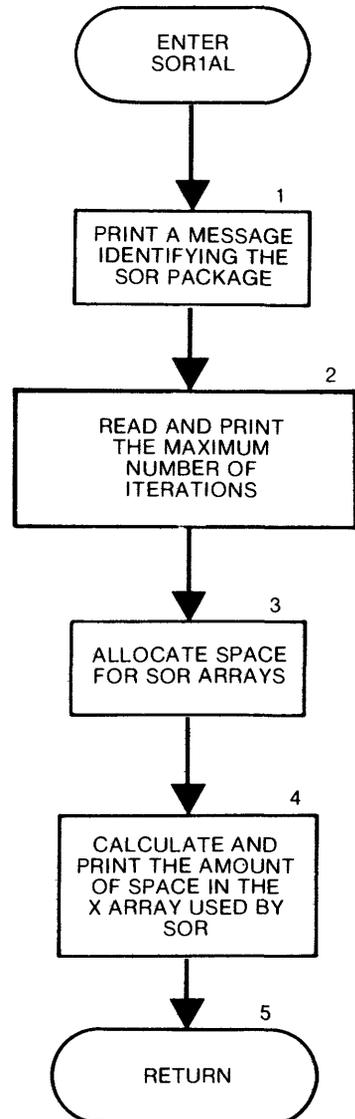
Module SORIAL allocates space in the X array for SOR arrays. The SOR arrays are A, RES, IEQPNT, HDCG, and LRCH. "A" holds the main diagonal and the lower diagonals of the symmetric coefficient matrix for a single slice. RES holds the residual vector (the right hand sides) for a single slice. IEQPNT holds a sequential identification number for each cell in a slice. HDCG holds the maximum head change for each iteration. LRCH holds the location of the cell (row, column, and layer) which had the maximum head change for each iteration.

Module SORIAL performs its functions in the following order:

1. Print a message identifying the SOR Package.
2. Read and print the maximum number of iterations.
3. Allocate the required space in the X array. The X-array location pointer (ISUM) is saved in the variable ISOLD prior to allocation so that the space required for SOR can be calculated in step 4. To allocate space for an array, the array-location variable is set equal to ISUM. Then ISUM is incremented by the required number of elements.
4. Calculate and print the space used in the X array. The space used by SOR is  $ISUM - ISOLD$ .
5. RETURN

## Flow Chart for Module SOR1AL

X array is the pool of memory space from which space is allocated for arrays used by various packages.



```

SUBROUTINE SORIAL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,LCIEQP,
1      MXITER,NCOL,NROW,NLAY,NSLICE,MBW,IN,IOUT)
C
C-----VERSION 1003 08DEC1983 SORIAL
C      *****
C      ALLOCATE STORAGE FOR SOR ARRAYS
C      *****
C
C      SPECIFICATIONS:
C      -----
C      -----
C
C1-----PRINT A MESSAGE IDENTIFYING SOR PACKAGE
      WRITE(IOUT,1)IN
      1 FORMAT(1H0,'SOR1 -- SLICE-SUCCESSIVE OVERRELAXATION PACKAGE'
      1,',',VERSION 1,12/08/83', ' INPUT READ FROM UNIT',I3)
C
C2-----READ AND PRINT MXITER (MAXIMUM # OF ITERATIONS)
      READ(IN,2) MXITER
      2 FORMAT(I10)
      WRITE(IOUT,3) MXITER
      3 FORMAT(1X,I5,' ITERATIONS ALLOWED FOR SOR CLOSURE')
C
C3-----ALLOCATE SPACE FOR THE SOR ARRAYS
      ISOLD=ISUM
      NSLICE=NCOL*NLAY
      MBW=NLAY+1
      LCA=ISUM
      ISUM=ISUM+NSLICE*MBW
      LCRES=ISUM
      ISUM=ISUM+NSLICE
      LCIEQP=ISUM
      ISUM=ISUM+NSLICE
      LCHDCG=ISUM
      ISUM=ISUM+MXITER
      LCLRCH=ISUM
      ISUM=ISUM+3*MXITER
      ISP=ISUM-ISOLD
C
C4-----CALCULATE AND PRINT THE SPACE USED IN THE X ARRAY
      WRITE(IOUT,4) ISP
      4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED BY SOR')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
      5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
      6 FORMAT(1X,' ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C5-----RETURN
      RETURN
      END

```

List of Variables for Module SORIAL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
IN	Package	Primary unit number from which input for this package will be read.
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.
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.
LCA	Package	Location in the X array of the first element of array A.
LCHDCG	Package	Location in the X array of the first element of array HDCG.
LCIEQP	Package	Location in the X array of the first element of array IEQPNT.
LCLRCH	Package	Location in the X array of the first element of array LRCH.
LCRES	Package	Location in the X array of the first element of array RES.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MBW	Package	Maximum bandwidth of the coefficient matrix +1.
MXITER	Package	Maximum number of iterations.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NSLICE	Package	Number of cells in a slice.

## Narrative for Module SOR1RP

Module SOR1RP reads data for the SOR package: the acceleration parameter (ACCL), also called the relaxation factor; the closure criterion (HCLOSE); and the time-step interval (IPRSOR) for printing head change. This module does not have a flow chart. Module SOR1RP performs its functions in the following order:

1. Read the acceleration parameter (ACCL), the closure criterion (HCLOSE), and the interval for printing head change (IPRSOR). If ACCL is zero, substitute a default value of 1.0. If IPRSOR is less than one, set it equal to 999.

2. Print the maximum number of iterations (MXITER), the acceleration parameter (ACCL), the closure criterion (HCLOSE), and the head-change interval (IPRSOR).

3. RETURN.

```

SUBROUTINE SOR1RP(MXITER,ACCL,HCLOSE,IN,IPRSOR,IOUT)
C
C
C-----VERSION 1005 16MAR1983 SOR1RP
C *****
C READ PARAMETERS FOR SOR
C *****
C
C SPECIFICATIONS:
C -----
C -----
C
C1-----READ THE ACCELERATION PARAMETER/RELAXATION FACTOR (ACCL) THE
C1-----CLOSURE CRITERION (HCLOSE) AND THE NUMBER OF TIME STEPS
C1-----BETWEEN PRINTOUTS OF MAXIMUM HEAD CHANGES (IPRSOR).
      READ(IN,1) ACCL,HCLOSE,IPRSOR
      1 FORMAT(2F10.0,I10)
      IF(ACCL.EQ.0.) ACCL=1.
      IF(IPRSOR.LT.1) IPRSOR=999
C
C2-----PRINT ACCL, HCLOSE, IPRSOR
      WRITE(IOUT,100)
      100 FORMAT(1H0,///57X,'SOLUTION BY SLICE-SUCCESSIVE OVERRELAXATION'
      1/57X,43('-'))
      WRITE(IOUT,115) MXITER
      115 FORMAT(1H0,47X,'MAXIMUM ITERATIONS ALLOWED FOR CLOSURE =',I9)
      WRITE(IOUT,120) ACCL
      120 FORMAT(1H ,63X,'ACCELERATION PARAMETER =',G15.5)
      WRITE(IOUT,125) HCLOSE
      125 FORMAT(1H ,52X,'HEAD CHANGE CRITERION FOR CLOSURE =',E15.5)
      WRITE(IOUT,130) IPRSOR
      130 FORMAT(1H ,52X,'SOR HEAD CHANGE PRINTOUT INTERVAL =',I9)
C
C3-----RETURN
      RETURN
      END

```

List of Variables for Module SOR1RP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ACCL	Package	Acceleration parameter.
HCLOSE	Package	Closure criterion for the iterative procedure.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRSOR	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
MXITER	Package	Maximum number of iterations.

### Narrative for Module SOR1AP

Module SOR1AP performs one iteration of the Slice-Successive Over-relaxation (SSOR) algorithm for solving the system of finite-difference equations. The conductances CC, CR, and CV and the composite terms HCOF and RHS (see equation 27) which are calculated by the formulation procedure are combined, row by row (slice by slice), to form the coefficient matrix  $\bar{A}$  and the vector  $\bar{RES}$  on the right hand side of the matrix equation for a single slice. Since the coefficient matrix is symmetric and banded, only main diagonals and NLAY subdiagonals are saved. As heads are calculated, they are stored in the array HNEW. The matrix  $\bar{A}$  and the vector  $\bar{RES}$  are passed to a submodule SSOR1B which solves the matrix equation for a vector of approximate head changes which is then multiplied by the relaxation factor to get the final head changes for the iteration. The final head changes are added to the heads from the preceding iteration to get the heads for the current iteration. The final head changes for the iteration are compared to the closure criterion to see if the iterative procedure has closed.

Module SOR1AP performs its functions in the following order:

1. Calculate the number of elements in the compressed coefficient matrix  $\bar{A}$ .
2. Process the slices (rows) one at a time (DO STEPS 3-7).
3. Clear the A array.
4. Assign integers sequentially to the active cells in the slice (remember that finite-difference equations are formulated only for active cells).

5. Calculate the elements in the compressed coefficient matrix  $\bar{A}$  and the residual vector  $\overline{RES}$ . Process the cells in the slice one cell at a time.

If the cell is inactive, move on to the next cell. The elements in the main diagonal of the coefficient matrix (the multipliers of  $h_{i,j,k}$ ) will consist of HCOF plus conductances to the six adjacent cells. They will be formed in an accumulator called EE. The contents of EE multiplied by the head from the previous iteration (HNEW) are subtracted from an accumulator (R) to form the residual.

(a) Determine the equation number (NEQ) of the cell. If NEQ is zero, the cell is inactive. Move on to the next cell.

(b) Set the accumulators EE and R equal to HCOF and RHS, respectively. Note: HNEW contains head from the last iteration.

(c) If there is a node to the left, subtract the conductance from EE and subtract the conductance times HNEW from R.

(d) If there is a node to the right, subtract the conductance from EE, and subtract the conductance times HNEW from R; and, if the cell to the right is active, move the conductance into the compressed coefficient matrix  $\bar{A}$ . Remember that the coefficient matrix is symmetric so the conductance to the left in step 5(c) did not have to be stored.

(e) If there is a node to the rear, subtract the conductance from EE and subtract the conductance times HNEW from R.

(f) If there is a node to the front, subtract the conductance from EE and subtract the conductance times HNEW from R. Remember that the

form of the SSOR equations does not have terms containing head in adjacent rows on the left hand side.

(g) If there is a node above, subtract the conductance from EE and subtract the conductance times HNEW from R.

(h) If there is a node below, subtract the conductance from EE and subtract the conductance times HNEW from R; and, if the cell below is active, move the conductance into A.

(i) Move EE into the first row of A. The first row in A corresponds to the main diagonal in the "full" coefficient matrix. Subtract EE times HNEW from R and store it in the residual vector.

6. If there are no equations for this slice, go on to the next slice. If there is only one equation, solve it directly and leave the result in the residual vector  $\overline{RES}$ . If there are two or more equations, call submodule SSOR1B to solve the system of equations for the slice leaving the results (first estimate of head change for this iteration) in the vector  $\overline{RES}$ .

7. For each cell in the slice, calculate the head for the current iteration.

(a) Multiply the first estimate of head change for this iteration by the relaxation factor to get the final estimate of head change for this iteration.

(b) Add the final head change for this iteration to the head from the last iteration to get the head for this iteration.

(c) If the head change for this cell is greater than that for any other cell, store the head change and the location of the cell.

8. Save the largest head change from this iteration so that it can be printed at the end of the time step.

9. Compare the biggest head change (BIGG) to the closure criterion (HCLOSE). If HCLOSE is greater than BIGG, set the convergence flag (ICNVG) equal to one.

10. If you have not converged and you have not exceeded the maximum number of iterations, RETURN.

11. Print the number of iterations.

12. If convergence failed, or this is the last time step, or this is the time step interval specified by the user, print the maximum head change for each iteration in this time step.

13. RETURN.

## Flow Chart for Module SOR1AP

A is a compressed coefficient matrix for a slice. It contains the main diagonal of the full matrix and the NLAY diagonals below it. (NLAY is the number of layers.)

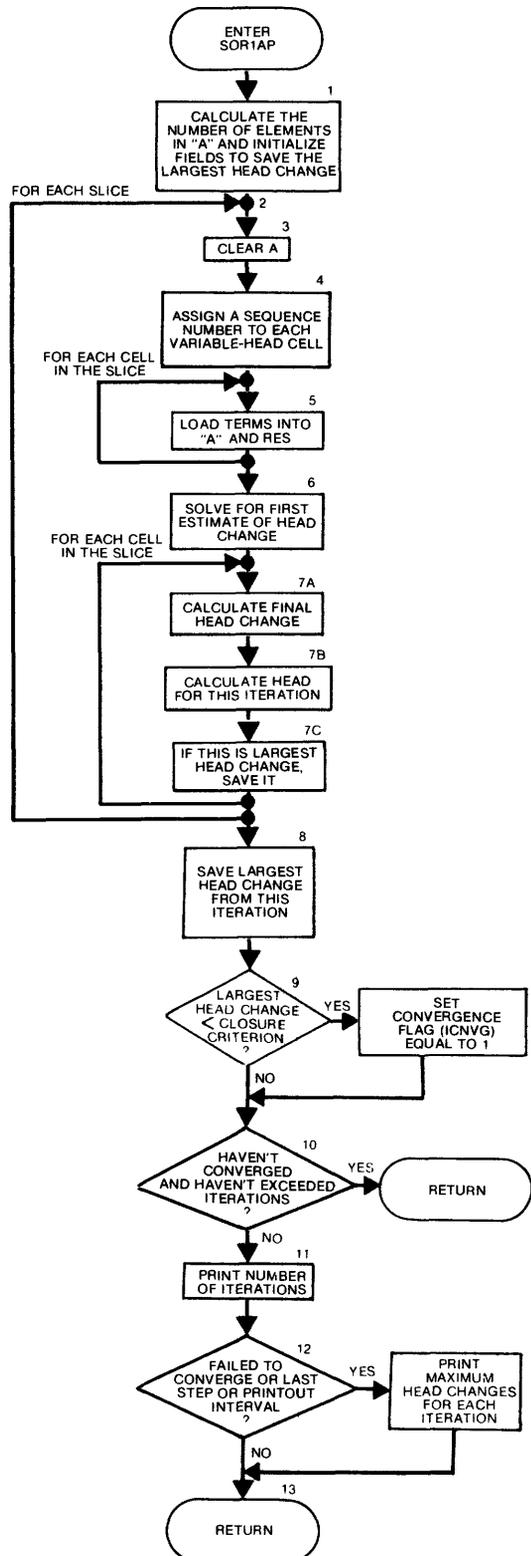
Sequence Number is a number used to identify the internal (variable-head) cells in a slice and also the equations for each internal cell.

RES is a vector containing the residuals for a slice. It consists of RHS (from the basic finite-difference equation) plus all of those terms which are moved to the right hand side to get the equations ready for solution in residual form.

First estimates of head change: these are the head changes calculated by simultaneously solving the equations for a slice. They will be multiplied by the relaxation factor to get final estimates of head change.

Final estimates of head change: these are the head changes calculated by multiplying first estimates by the relaxation factor. They are added to the heads from the previous iteration to get head for the current iteration.

ICNVG is the convergence flag. It is set in the approximator and returned to the MAIN Program so that the iteration loop can be terminated.



```

SUBROUTINE SORIAP(HNEW,IBOUND,CR,CC,CV,HCOF,RHS,A,RES,IEQPNT,
1   HDCG,LRCH,KITER,HCLOSE,ACCL,ICNVG,KSTP,KPER,
2   IPRSOR,MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C-----VERSION 0936 09MAY1983 SORIAP
C *****
C SOLUTION BY SLICE-SUCCESSIVE OVERRELAXATION -- 1 ITERATION
C *****
C
C SPECIFICATIONS:
C -----
C DOUBLE PRECISION HNEW,DIFF,DP,EE,R,HCFHNW,HHCOF
C
C DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
1   CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
1   CV(NCOL,NROW,NLAY), HCOF(NCOL,NROW,NLAY), RHS(NCOL,NROW,NLAY),
2   HDCG(MXITER), LRCH(3,MXITER),A(MBW,NSLICE),RES(NSLICE),
3   IEQPNT(NLAY,NCOL)
C -----
C
C1-----CALCULATE # OF ELEMENTS IN COMPRESSED MATRIX A AND
C1-----INITIALIZE FIELDS TO SAVE LARGEST HEAD CHANGE.
      NA=MBW*NSLICE
      BIG=0.
      ABSBIG=0.
      IB=0
      JB=0
      KB=0
C
C2-----PROCESS EACH SLICE
      DO 500 I=1,NROW
C
C3-----CLEAR A
      DO 110 J=1,NSLICE
      DO 110 K=1,MBW
110 A(K,J)=0.
C
C4-----ASSIGN A SEQUENCE # TO EACH VARIABLE HEAD CELL.
      NEQT=0
      DO 200 J=1,NCOL
      DO 200 K=1,NLAY
      IEQPNT(K,J)=0
      IF(IBOUND(J,I,K).LE.0) GO TO 200
      NEQT=NEQT+1
      IEQPNT(K,J)=NEQT
200 CONTINUE
C
C5-----FOR EACH CELL LOAD MATRIX A AND VECTOR RES
      DO 300 J=1,NCOL
      DO 300 K=1,NLAY
C
C5A-----IF SEQUENCE # IS 0 (CELL IS EXTERNAL) GO ON TO NEXT CELL
      NEQ=IEQPNT(K,J)
      IF(NEQ.EQ.0) GO TO 300
C
C5B-----INITIALIZE ACCUMULATORS EE AND R
      EE=0.
      R=RHS(J,I,K)
C
C5C-----IF NODE TO LEFT SUBTRACT TERMS FROM EE AND R
      IF(J.EQ.1) GO TO 120

```

```

      DP=CR(J-1,I,K)
      R=R-DP*HNEW(J-1,I,K)
      EE=EE-DP
C
C5D-----IF NODE TO RIGHT SUBTRACT TERMS FROM EE & R, MOVE COND TO A
120 IF(J,EQ.NCOL) GO TO 125
      SP=CR(J,I,K)
      DP=SP
      R=R-DP*HNEW(J+1,I,K)
      EE=EE-DP
      NXT=IEQPNT(K,J+1)
      IF(NXT.GT.0) A(1+NXT-NEQ,NEQ)=SP
C
C5E-----IF NODE TO REAR SUBTRACT TERMS FROM EE AND R
125 IF(I,EQ.1) GO TO 130
      DP=CC(J,I-1,K)
      R=R-DP*HNEW(J,I-1,K)
      EE=EE-DP
C
C5F-----IF NODE TO FRONT SUBTRACT TERMS FROM EE AND R
130 IF(I,EQ.NROW) GO TO 132
      DP=CC(J,I,K)
      R=R-DP*HNEW(J,I+1,K)
      EE=EE-DP
C
C5G-----IF NODE ABOVE SUBTRACT TERMS FROM EE AND R
132 IF(K,EQ.1) GO TO 134
      DP=CV(J,I,K-1)
      R=R-DP*HNEW(J,I,K-1)
      EE=EE-DP
C
C5H-----IF NODE BELOW SUBTRACT TERMS FROM EE & R AND MOVE COND TO A
134 IF(K,EQ.NLAY) GO TO 136
      SP=CV(J,I,K)
      DP=SP
      R=R-DP*HNEW(J,I,K+1)
      EE=EE-DP
      IF(IEQPNT(K+1,J).GT.0) A(2,NEQ)=SP
C
C5I-----MOVE EE INTO A, SUBTRACT EE TIMES LAST HEAD FROM R TO GET RES
136 HHCOF=HCOF(J,I,K)
      A(1,NEQ)=EE+HHCOF
      HNW=HNEW(J,I,K)
      HCFHNW=HNW*HCOF(J,I,K)
      RES(NEQ)=R-EE*HNEW(J,I,K)-HCFHNW
300 CONTINUE
C
C6-----IF NO EQUATIONS GO TO NEXT SLICE, IF ONE EQUATION SOLVE
C6-----DIRECTLY, IF 2 EQUATIONS CALL SSOR1B TO SOLVE FOR FIRST
C6-----ESTIMATE OF HEAD CHANGE FOR THIS ITERATION.
      IF(NEQT.LT.1) GO TO 500
      IF(NEQT.EQ.1) RES(1)=RES(1)/A(1,1)
      IF(NEQT.GE.2) CALL SSOR1B(A,RES,NEQT,NA,MBW)
C
C7-----FOR EACH CELL IN SLICE CALCULATE FINAL HEAD CHANGE THEN HEAD.
      DO 400 J=1,NCDL
      DO 400 K=1,NLAY
      NEQ=IEQPNT(K,J)
      IF(NEQ.EQ.0) GO TO 400
C
C7A-----MULTIPLY FIRST ESTIMATE OF HEAD CHANGE BY RELAX FACTOR TO

```

```

C7A-----GET FINAL ESTIMATE OF HEAD CHANGE FOR THIS ITERATION.
      DH=RES(NEQ)*ACCL
      DIFF=DH
C
C7B-----ADD FINAL ESTIMATE TO HEAD FROM LAST ITERATION TO GET HEAD
C7B-----FOR THIS ITERATION.
      HNEW(J,I,K)=HNEW(J,I,K)+DIFF
C
C7C-----SAVE FINAL HEAD CHANGE IF IT IS THE LARGEST
      ABSDH=ABS(DH)
      IF(ABSDH.LE.ABSBIG) GO TO 400
      ABSBIG=ABSDH
      BIG=DH
      IB=I
      JB=J
      KB=K
      400 CONTINUE
C
C
      500 CONTINUE
C
C8-----SAVE LARGEST HEAD CHANGE FOR THIS ITERATION
      HDCG(KITER)=BIG
      LRCH(1,KITER)=KB
      LRCH(2,KITER)=IB
      LRCH(3,KITER)=JB
C
C9-----IF LARGEST HEAD CHANGE IS SMALLER THAN CLOSURE THEN SET
C9-----CONVERGE FLAG (ICNVG) EQUAL TO 1.
      ICNVG=0
      IF(ABSBIG.LE.HCLOSE) ICNVG=1
C
C10-----IF NOT CONVERGED AND NOT EXCEEDED ITERATIONS THEN RETURN
      IF(ICNVG.EQ.0 .AND. KITER.NE.MXITER) RETURN
      IF(KSTP.EQ.1) WRITE(IOUT,600)
      600 FORMAT(1H0)
C
C11-----PRINT NUMBER OF ITERATIONS
      WRITE(IOUT,601) KITER,KSTP,KPER
      601 FORMAT(1X,I5,' ITERATIONS FOR TIME STEP',I4,' IN STRESS PERIOD',
1          13)
C
C12-----IF FAILED TO CONVERGE OR LAST TIME STEP OR PRINTOUT
C12-----INTERVAL SPECIFIED BY USER IS HERE THEN PRINT MAXIMUM
C12-----HEAD CHANGES FOR EACH ITERATION.
      IF(ICNVG.NE.0 .AND. KSTP.NE.NSTP .AND. MOD(KSTP,IPRSOR).NE.0)
1          GO TO 700
      WRITE(IOUT,5)
      5 FORMAT(1H0,'MAXIMUM HEAD CHANGE FOR EACH ITERATION:/'
1          1H0,4(' HEAD CHANGE LAYER,ROW,COL')/1X,120('-'))
      WRITE (IOUT,10) (HDCG(J),(LRCH(I,J),I=1,3),J=1,KITER)
      10 FORMAT(1X,4(4X,G12.4,' (' ,I3,' ,',I3,' ,',I3,' ')))
      WRITE(IOUT,11)
      11 FORMAT(1H0)
C
C13-----RETURN
      700 RETURN
C
      END

```

List of Variables for Module SOR1AP

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
A	Package	DIMENSION (MBW,NSLICE), Compressed coefficient matrix for a slice.
ABSDIG	Module	Largest ABSDH for this iteration.
ABSDH	Module	Absolute value of head change in a cell for the current iteration.
ACCL	Package	Acceleration parameter.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DH	Module	Change in head in a cell during one iteration.
DIFF	Module	Double-precision change in head (DH).
DP	Module	Double-precision temporary field.
EE	Module	Main diagonal term in the finite-difference equation.
HCLOSE	Package	Closure criterion for the iterative procedure.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HDCG	Package	DIMENSION (MXITER), Maximum head change for each iteration.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
I	Module	Index for rows.
IB	Module	Row number of the cell containing the largest head change.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
ICNVG	Global	Flag is set equal to one when the iteration procedure has converged.
IEQPNT	Global	DIMENSION (NLAY,NCOL), Sequence numbers for variable-head cells in a slice.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRSOR	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
J	Module	Index for columns.
JB	Module	Column number of the cell containing the largest head change.
K	Module	Index for layers.
KB	Module	Layer number of the cell containing the largest head change.
KITER	Global	Iteration counter. Reset at the start of each time step.
KPER	Global	Stress period counter.

List of Variables for Module SOR1AP (Continued)

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
KSTP	Global	Time step counter. Reset at the start of each stress period.
LRCH	Package	DIMENSION (MXITER), Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MBW	Package	Maximum bandwidth of the coefficient matrix +1.
MXITER	Package	Maximum number of iterations.
NA	Package	Number of elements in the compressed coefficient matrix (A).
NCOL	Global	Number of columns in the grid.
NEQ	Module	Index for equations (variable-head cells) in a slice.
NEQT	Package	Number of equations (variable-head cells) in a slice.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NSLICE	Package	Number of cells in a slice.
NSTP	Global	Number of time steps in the current stress period.
NXT	Module	Sequence number of the cell to the right.
R	Module	Right hand side of the finite-difference equation as modified (terms for the adjacent rows moved to the right) for solution by the slice-successive overrelaxation.
RES	Package	DIMENSION (NSLICE), Residual.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.
SP	Module	Single-precision temporary field.

## Narrative for Module SSOR1B

Module SSOR1B uses Gaussian elimination to solve a matrix equation of the form  $\bar{A}\bar{X} = \bar{B}$ . The matrix  $\bar{A}$  is symmetric and banded with rank "N." It is stored in the compressed format shown in figure 55 and uses a single subscript. The vector  $\bar{X}$  as it is calculated is stored in the space reserved for vector  $\bar{B}$ .

The indices used in the module flow chart are those for standard matrix organization. The indices actually used in the program are based on the compressed format and a single index. Module SSOR1B performs its functions in the following order:

1. Work through the first N-1 rows using each one, in sequence, as the pivot row (row I).
2. Calculate the inverse of the main diagonal element--- $a_{I,I}$ . (The index (ID) points to  $a_{I,I}$  which is the first element of column I in the compressed matrix  $\bar{A}$ .)
3. Modify each of the rows after row I so that the terms corresponding to the pivot term are eliminated. Since the coefficient matrix is banded, there are only MBW-1 equations (where MBW is the maximum half-bandwidth plus one) where the term to be eliminated is not already equal to zero. The rows are indexed by "L." (The equation corresponding to row "L" is referred to as equation "L.")
4. Calculate the coefficient C which when multiplied by the pivot equation and subtracted from equation L will eliminate a term in equation L.

(The index IB points to a coefficient in the pivot equation which, because of symmetry, is equal to the coefficient to be eliminated.)

5. Calculate the new coefficients in equation L for each of the terms to the right of the coefficient that is being eliminated. Because the matrix is banded, there are only MBW-1 nonzero terms to the right of the pivot. Therefore, at most, MBW-1 coefficients have to be calculated.

6. Subtract C times a coefficient in the pivot equation from the corresponding coefficient in equation L.

7. Subtract C times the right side of the pivot equation from the right side of equation L. (The index LB points to the coefficient in equation L which must be calculated.)

8. Solve equation N for X(N) putting the result in B(N).

9. Work backward from equation N-1 solving each equation (equation L) for X(L).

10. Set the accumulator "SUM" equal to zero.

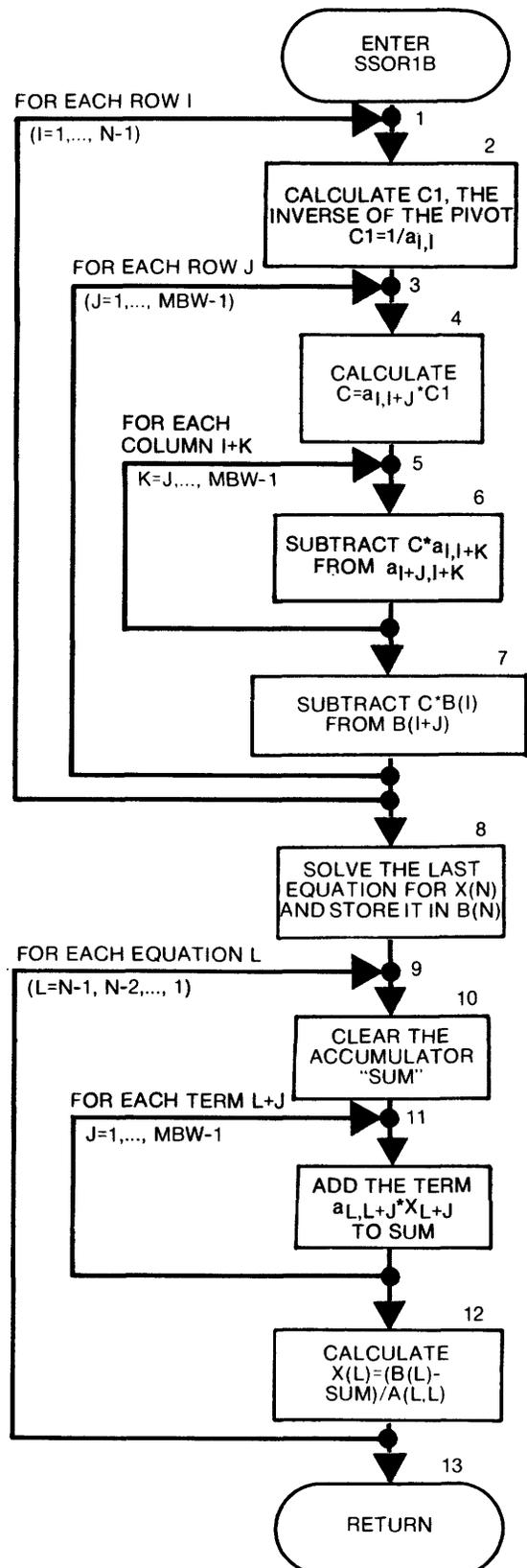
11. Multiply the coefficient to the right of the main diagonal (in equation L) by the corresponding value of X and add it to the sum.

12. Calculate the value of X(L) and store it in B(L).

13. RETURN.

## Flow Chart for Module SSOR1B

MBW is the maximum half bandwidth of the coefficient matrix plus one.



```

SUBROUTINE SSOR1B(A,B,N,NA,MBW)
C
C
C-----VERSION 1359 31MAR1983 SSOR1B
C *****
C SOLVE A SYMMETRIC SET OF EQUATIONS
C   A IS COEFFICIENT MATRIX IN COMPRESSED FORM
C   B IS RIGHT HAND SIDE AND IS REPLACED BY SOLUTION
C   N IS NUMBER OF EQUATIONS TO BE SOLVED
C   MBW IS BANDWIDTH OF A
C   NA IS ONE-DIMENSION SIZE OF A
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION A(NA),B(N)
C -----
C
C   NM1=N-1
C   MBW1=MBW-1
C   ID=1-MBW
C
C1-----SEQUENTIALLY USE EACH OF THE FIRST N-1 ROWS AS
C1-----THE PIVOT ROW.
C   DO 20 I=1,NM1
C
C2-----CALCULATE THE INVERSE OF THE PIVOT.
C   ID=ID+MBW
C   C1=1./A(ID)
C   LD=ID
C   L=I
C
C3-----FOR EACH ROW AFTER THE PIVOT ROW (THE TARGET ROW)
C3-----ELIMINATE THE COLUMN CORRESPONDING TO THE PIVOT.
C   DO 15 J=1,MBW1
C   L=L+1
C   IF(L.GT.N) GO TO 20
C   IB=ID+J
C
C4-----CALCULATE THE FACTOR NEEDED TO ELIMINATE A TERM IN THE
C4-----TARGET ROW.
C   C=A(IB)*C1
C   LD=LD+MBW
C   LB=LD-1
C
C5-----MODIFY THE REST OF THE TERMS IN THE TARGET ROW.
C   DO 10 K=J,MBW1
C
C6-----SUBTRACT THE FACTOR TIMES A TERM IN THE PIVOT ROW
C6-----FROM THE CORRESPONDING COLUMN IN THE TARGET ROW.
C   LB=LB+1
C   A(LB)=A(LB)-C*A(ID+K)
C   10 CONTINUE

```

```

C
C7-----MODIFY THE RIGHT SIDE OF THE EQUATION CORRESPONDING
C7-----TO THE TARGET ROW.
      B(I+J)=B(I+J)-C*B(I)
      15 CONTINUE
      20 CONTINUE
      ID=ID+MBW
C
C8-----SOLVE THE LAST EQUATION.
      B(N)=B(N)/A(ID)
C
C9-----WORKING BACKWARDS SOLVE THE REST OF THE EQUATIONS.
      DO 70 I=1,NM1
      ID=ID-MBW
C
C10-----CLEAR THE ACCUMULATOR SUM.
      SUM=0.0
      L=N-I
      MBW1M=MINO(MBW1,I)
C
C11-----ADD THE KNOWN TERMS IN EQUATION L TO SUM.
      DO 60 J=1,MBW1M
      SUM=SUM+A(ID+J)*B(L+J)
      60 CONTINUE
C
C12-----SOLVE FOR THE ONE UNKNOWN IN EQUATION L.
      B(L)=(B(L)-SUM)/A(ID)
      70 CONTINUE
C
C13-----RETURN
      RETURN
      END

```

List of Variables for Module SSOR1B

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
A	Package	DIMENSION (MBW,NSLICE), Compressed coefficient matrix for a slice.
B	Package	DIMENSION (N), Right-hand-side vector.
C	Module	Factor needed to eliminate a term in the target row.
C1	Module	Inverse of pivot.
I	Module	Index for rows in the SSOR matrix (not the grid).
IB	Module	Index for elements to the right of the pivot.
ID	Module	Index of pivots.
J	Module	Index for columns.
K	Module	Index for columns.
L	Module	Index for equations.
LB	Module	Index for elements in the target row to the right of the main diagonal.
LD	Module	Index of the main diagonal elements in the target rows.
MBW	Package	Maximum bandwidth of the coefficient matrix +1.
MBW1	Module	Maximum bandwidth of the coefficient matrix.
MBW1M	Module	Maximum possible number of the nonzero elements to the right of the main diagonal.
N	Package	Number of equations to be solved.
NA	Package	One-dimension size of compressed matrix "A."
NM1	Module	N-1.
SUM	Module	In back substitution--in equation L, sum of terms to the right of the main diagonal term (L,L).

CHAPTER 14  
UTILITY MODULES

Utility modules are those submodules which perform general tasks common to several different packages. The name of a utility module is a "U" followed by a five-character mnemonic. There are eight utility modules:

UBUDSV	Writes an unformatted record consisting of an array with one real number for each cell in the grid.
ULASAV	Writes an unformatted record consisting of an array with one real number for each cell in a layer.
ULAPRS and ULAPRW	Prints one two-dimensional array which contains one real number for each cell in a layer. ULAPRS prints, in strip form, the first N columns (where N is the number of values that can fit on one print line) of each row and then the next N columns, etc., until all columns of each row are printed (fig. 56). ULAPRW prints, in wrap form, all of row 1, all of row 2, and all of row 3, etc. The format for printing arrays is shown in table 2.
UCOLNO	Prints column numbers at the top of each page of data printed by ULAPRS and ULAPRW.
U2DREL	Reads a two-dimensional array of real numbers.
U2DINT	Reads a two-dimensional array of integers.
U1DREL	Reads a one-dimensional array of real numbers.

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17			
1	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
2	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
3	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
4	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
5	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
6	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
7	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			

WRAP FORM

	1	2	3	4	5	6	7	8	9	10
1	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
2	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
3	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
4	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
5	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
6	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
7	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79

	11	12	13	14	15	16	17
1	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
2	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
3	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
4	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
5	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
6	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
7	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79

STRIP FORM

Figure 56.—Illustration of wrap and strip forms of printed output for a layer containing 7 rows and 17 columns.

Table 2.--Print-format codes for utility modules  
ULAPRS and ULAPRW.

<u>IPRN</u>	<u>FORMAT</u>
1	11G10.3
2	9G13.6
3	15F7.1
4	15F7.2
5	15F7.3
6	15F7.4
7	20F5.0
8	20F5.1
9	20F5.2
10	20F5.3
11	20F5.4
12	10G11.4

## Utility Input

The utility module only reads from channels specified by calling programs. The real two-dimensional array reader (U2DREL), the integer two-dimensional array reader (U2DINT), and the real one-dimensional array reader (U1DREL) read one array-control record and, optionally, a data array in a format specified on the array-control record.

FOR REAL ARRAY READER (U2DREL or U1DREL)

Data:	LOCAT	CNSTNT	FMTIN	IPRN
Format:	I10	F10.0	5A4	I10

FOR INTEGER ARRAY READER (U2DINT)

Data:	LOCAT	ICONST	FMTIN	IPRN
Format:	I10	I10	5A4	I10

### Explanation of Fields Used in Input Instructions

LOCAT--indicates the location of the data which will be put in the array.

If LOCAT < 0, the sign is reversed to give the unit number from which an unformatted record will be read.

If LOCAT = 0, every element in the array will be set equal to the value CNSTNT/ICONST.

If LOCAT > 0, it is the unit number from which data values will be read in the format specified in the third field of the array-control record (FMTIN).

CNSTNT/ICONST--is a constant. Its use depends on the value of LOCAT.

If LOCAT = 0, every element in the array is set equal to CNSTNT/ICONST.

If LOCAT ≠ 0, and if CNSTNT/ICONST ≠ 0, every element in the array is multiplied by CNSTNT/ICONST.

FMTIN--is the format of records containing the array values. It is used only if the first field in the array-control record (LOCAT) contains a positive number. The format must be enclosed in parentheses; for example, (15F5.0) for real data and (15I5) for integer data.

IPRN--is a flag indicating that the array being read should be printed and a code for indicating the format that should be used. It is used only if LOCAT is not equal to zero. The format codes are different for each

of the three modules. IPRN is set to zero when the specified value exceeds those defined in the chart below. If IPRN is less than zero, the array will not be printed.

<u>IPRN</u>	<u>U2DREL</u>	<u>U2DINT</u>	<u>U1DREL</u>
0	10G11.4	10I11	10G12.5
1	11G10.3	60I1	
2	9G13.6	40I2	
3	15F7.1	30I3	
4	15F7.2	25I4	
5	15F7.3	20I5	
6	15F7.4		
7	20F5.0		
8	20F5.1		
9	20F5.2		
10	20F5.3		
11	20F5.4		
12	10G11.4		

### Narrative for Module UBUDSV

Utility module UBUDSV writes an unformatted record consisting of an array dimensioned (NCOL, NROW, NLAY). The record containing the array is preceded by an unformatted record containing identifying information. The identifying information consists of:

KSTP	current time step	integer	1 word
KPER	current stress period	integer	1 word
TEXT	label	character string	4 words
NCOL	number of columns	integer	1 word
NROW	number of rows	integer	1 word
NLAY	number of layers	integer	1 word

Documentation of this module consists only of comments in the program and a list of variables.

```

SUBROUTINE UBUDSV(KSTP,KPER,TEXT,IBDCHN,BUFF,NCOL,NROW,NLAY,IOUT)
C
C
C-----VERSION 1305 28DEC1983 UBUDSV
C*****
C RECORD CELL-BY-CELL FLOW TERMS FOR ONE COMPONENT OF FLOW.
C*****
C
C SPECIFICATIONS:
C-----
C DIMENSION TEXT(4),BUFF(NCOL,NROW,NLAY)
C-----
C
C1-----WRITE AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1-----INFORMATION.
      WRITE(IOUT,1) TEXT,IBDCHN,KSTP,KPER
      1 FORMAT(1X,'" ',4A4,'" BUDGET VALUES WILL BE SAVED ON UNIT',I3,
      1      ' AT END OF TIME STEP',I3,' , STRESS PERIOD',I3)
C
      WRITE(IBDCHN) KSTP,KPER,TEXT,NCOL,NROW,NLAY
C
C2-----WRITE AN UNFORMATTED RECORD CONTAINING VALUES FOR
C2-----EACH CELL IN THE GRID. THE ARRAY IS DIMENSIONED
C2----- (NCOL,NROW,NLAY)
      WRITE(IBDCHN) BUFF
C
C3-----RETURN
      RETURN
      END

```

List of Variables for Module UBUDSV

<u>Variation</u>	<u>Range</u>	<u>Definition</u>
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
IBDCHN	Module	Unit number on which the array will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
KPER	Global	Stress period counter.
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.
TEXT	Module	Label to be printed or recorded with the array data.

### Narrative for Module ULASAV

Utility module ULASAV writes an unformatted record consisting of an array dimensioned (NCOL, NROW). The record containing the array is preceded by an unformatted record containing identifying information. The identifying information consists of:

KSTP	current time step	integer	1 word
KPER	current stress period	integer	1 word
PERTIM	elapsed time in the current stress period	real	1 word
TOTIM	elapsed time in the simulation	real	1 word
TEXT	label	character string	4 words
NCOL	number of columns	integer	1 word
NROW	number of rows	integer	1 word
ILAY	layer number	integer	1 word

```

SUBROUTINE ULASAV(BUF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,
1          NROW,ILAY,ICHN)
C
C-----VERSION 1445 20APR1983 ULASAV
C *****
C SAVE 1 LAYER ARRAY ON DISK
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION BUF(NCOL,NROW),TEXT(4)
C -----
C
C1-----WRITE AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1-----INFORMATION.
C WRITE(ICHN) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY
C
C2-----WRITE AN UNFORMATTED RECORD CONTAINING ARRAY VALUES
C2-----THE ARRAY IS DIMENSIONED (NCOL,NROW)
C WRITE(ICHN) ((BUF(IC,IR),IC=1,NCOL),IR=1,NROW)
C
C3-----RETURN
C RETURN
C END

```

List of Variables for Module ULASAV

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUF	Module	Buffer containing data to be printed or recorded.
IC	Module	Index for columns.
ICHN	Module	Unit number on which the array is to be recorded.
ILAY	Module	Layer number.
IR	Module	Index for rows.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NROW	Global	Number of rows in the grid.
PERTIM	Package	Elapsed time during the current stress period.
TEXT	Module	Label to be printed or recorded with the array data.
TOTIM	Package	Elapsed time in the simulation.

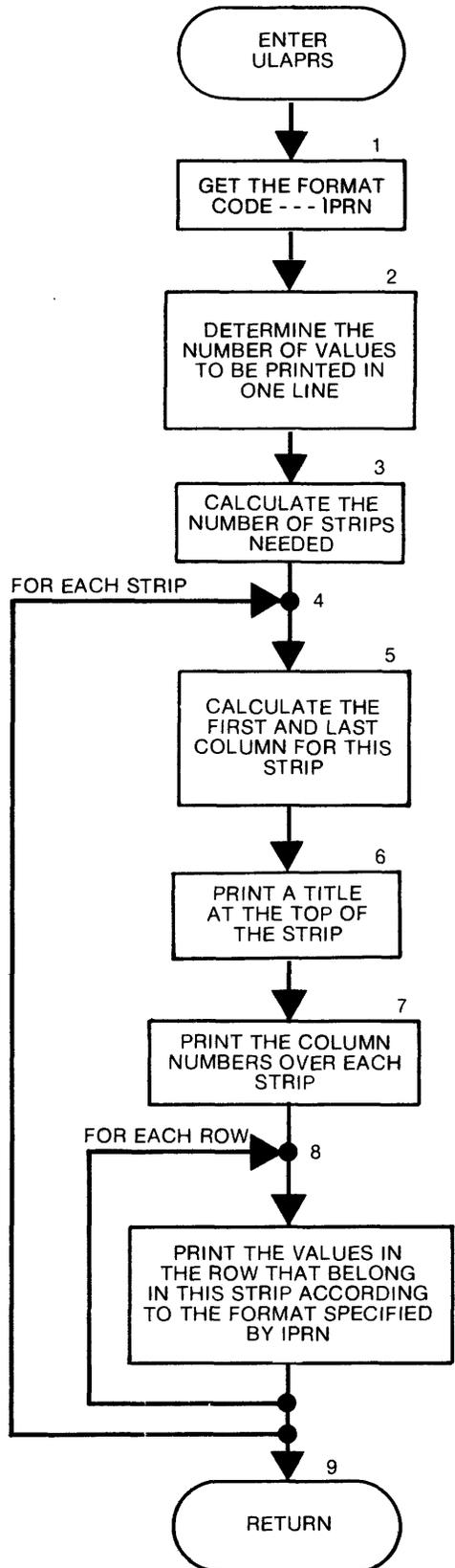
### Narrative for Module ULAPRS

Module ULAPRS prints a two-dimensional array in strip form (fig. 56) using one of twelve FORTRAN formats. Module ULAPRS performs its tasks in the following order:

1. Get the format code (IP). If it is less than 1 or greater than 12, set it equal to 12 (the default).
2. Use the format code (IP) to determine the number of values (NCAP) to be printed on one line.
3. Calculate the number of spaces used for each value (NCPF) and the number of strips (NSTRIP). Initialize the fields to store the first column (J1) and the last column (J2) for each strip.
4. Loop through the strips (DO STEPS 5-8).
5. Calculate the first (J1) and last (J2) column for this strip.
6. Print a title on each strip.
7. Call module UCOLNO to print the column numbers above each strip.
8. Loop through the rows printing columns J1 through J2 using the appropriate format (IP).
9. RETURN.

Flow Chart for Module ULAPRS

IPRN is a code indicating the format to be used in printing array values. If it is not between 1 and 12, it is set equal to 12.



```

SUBROUTINE ULAPRS(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,IPRN,IOUT)
C
C
C-----VERSION 1448 20APR1983 ULAPRS
C *****
C PRINT A 1 LAYER ARRAY IN STRIPS
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION BUF(NCOL,NROW),TEXT(4)
C -----
C
C1-----MAKE SURE THE FORMAT CODE (IP OR IPRN) IS BETWEEN 1
C1-----AND 12.
C IP=IPRN
C IF(IP.LT.1 .OR. IP.GT.12) IP=12
C
C2-----DETERMINE THE NUMBER OF VALUES (NCAP) PRINTED ON ONE LINE.
C IF(IP.EQ.1) NCAP=11
C IF(IP.EQ.2) NCAP=9
C IF(IP.GT.2 .AND. IP.LT.7) NCAP=15
C IF(IP.GT.6 .AND. IP.LT.12) NCAP=20
C IF(IP.EQ.12) NCAP=10
C
C3-----CALCULATE THE NUMBER OF STRIPS (NSTRIP).
C NCPF=129/NCAP
C ISP=0
C IF(NCAP.GT.12) ISP=3
C NSTRIP=(NCOL-1)/NCAP + 1
C J1=1-NCAP
C J2=0
C
C4-----LOOP THROUGH THE STRIPS.
C DO 2000 N=1,NSTRIP
C
C5-----CALCULATE THE FIRST(J1) & THE LAST(J2) COLUMNS FOR THIS STRIP
C J1=J1+NCAP
C J2=J2+NCAP
C IF(J2.GT.NCOL) J2=NCOL
C
C6-----PRINT TITLE ON EACH STRIP
C WRITE(IOUT,1) TEXT,ILAY,KSTP,KPER
C 1 FORMAT(1H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
C 1 ' IN STRESS PERIOD',I3/11X,7I('-'))
C
C7-----PRINT COLUMN NUMBERS ABOVE THE STRIP
C CALL UCOLNO(J1,J2,ISP,NCAP,NCPF,IOUT)
C
C8-----LOOP THROUGH THE ROWS PRINTING COLS J1 THRU J2 WITH FORMAT IP
C DO 1000 I=1,NROW
C GO TO(10,20,30,40,50,60,70,80,90,100,110,120), IP
C
C-----FORMAT 10G10.3
C 10 WRITE(IOUT,11) I,(BUF(J,I),J=J1,J2)
C 11 FORMAT(1H0,I3,2X,1PG10.3,10(1X,G10.3))
C GO TO 1000
C
C-----FORMAT 8G13.6
C 20 WRITE(IOUT,21) I,(BUF(J,I),J=J1,J2)
C 21 FORMAT(1H0,I3,2X,1PG13.6,8(1X,G13.6))
C GO TO 1000
C
C-----FORMAT 15F7.1
C 30 WRITE(IOUT,31) I,(BUF(J,I),J=J1,J2)
C 31 FORMAT(1H0,I3,1X,15(1X,F7.1))
C GO TO 1000
C
C-----FORMAT 15F7.2
C 40 WRITE(IOUT,41) I,(BUF(J,I),J=J1,J2)
C 41 FORMAT(1H0,I3,1X,15(1X,F7.2))
C GO TO 1000
C
C-----FORMAT 15F7.3
C 50 WRITE(IOUT,51) I,(BUF(J,I),J=J1,J2)
C 51 FORMAT(1H0,I3,1X,15(1X,F7.3))
C GO TO 1000

```

```

C
C-----FORMAT 15F7.4
  60 WRITE(IOUT,61) I,(BUF(J,I),J=J1,J2)
  61 FORMAT(1H0,I3,1X,15(1X,F7.4))
     GO TO 1000
C
C-----FORMAT 20F5.0
  70 WRITE(IOUT,71) I,(BUF(J,I),J=J1,J2)
  71 FORMAT(1H0,I3,1X,20(1X,F5.0))
     GO TO 1000
C
C-----FORMAT 20F5.1
  80 WRITE(IOUT,81) I,(BUF(J,I),J=J1,J2)
  81 FORMAT(1H0,I3,1X,20(1X,F5.1))
     GO TO 1000
C
C-----FORMAT 20F5.2
  90 WRITE(IOUT,91) I,(BUF(J,I),J=J1,J2)
  91 FORMAT(1H0,I3,1X,20(1X,F5.2))
     GO TO 1000
C
C-----FORMAT 20F5.3
 100 WRITE(IOUT,101) I,(BUF(J,I),J=J1,J2)
 101 FORMAT(1H0,I3,1X,20(1X,F5.3))
     GO TO 1000
C
C-----FORMAT 20F5.4
 110 WRITE(IOUT,111) I,(BUF(J,I),J=J1,J2)
 111 FORMAT(1H0,I3,1X,20(1X,F5.4))
     GO TO 1000
C
C-----FORMAT 9G11.4
 120 WRITE(IOUT,121) I,(BUF(J,I),J=J1,J2)
 121 FORMAT(1H0,I3,2X,1PG11.4,9(1X,G11.4))
C
 1000 CONTINUE
 2000 CONTINUE
C
C9-----RETURN
      RETURN
      END

```

### List of Variables for Module ULAPRS

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUF	Module	Buffer containing data to be printed or recorded.
I	Module	Index for rows.
ILAY	Module	Layer number.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IP	Module	Format code.
IPRN	Module	Code for the format to be used when printing arrays.
ISP	Module	Number of spaces.
J	Module	Index for columns.
J1	Module	First column in a strip.
J2	Module	Last column in a strip.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
N	Module	Index for strips.
NCAP	Module	Number of columns on a line.
NCOL	Global	Number of columns in the grid.
NCPF	Module	Number of columns per field.
NROW	Global	Number of rows in the grid.
NSTRIP	Module	Number of strips.
TEXT	Module	Label to be printed or recorded with the array data.

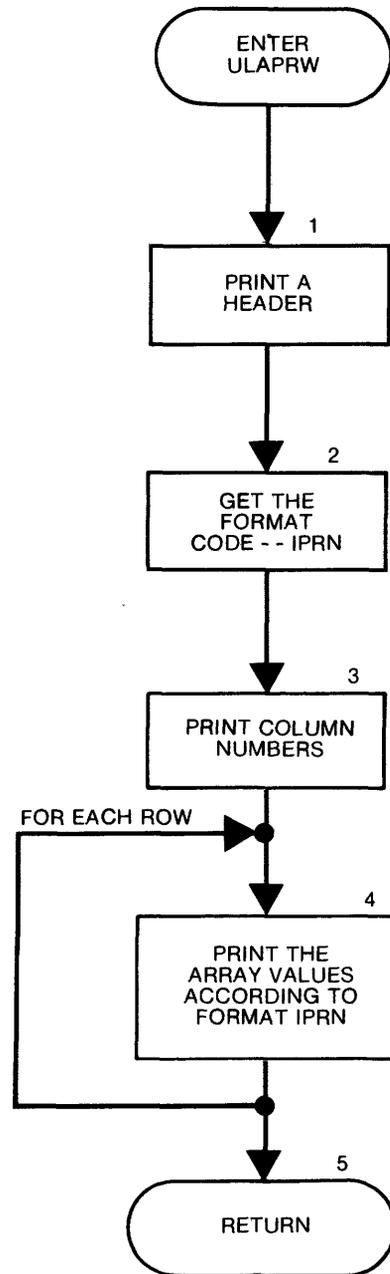
### Narrative for Module ULAPRW

Module ULAPRW prints a two-dimensional array in wrap form (fig. 56) using one of twelve FORTRAN formats. Module ULAPRW performs its tasks in the following order:

1. Print a header.
2. Set the format code (IP). If it is less than 1 or greater than 12, set it equal to 12 (the default).
3. Call the module UCOLNO to print column numbers.
4. Loop through the rows printing each one in its entirety using the appropriate format code.
5. RETURN.

Flow Chart for Module ULAPRW

IPRN is a code indicating the format to be used in printing array values. If it is not between 1 and 12, it is set equal to 12.



```

SUBROUTINE ULAPRW(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,I PRN,IOUT)
C
C
C-----VERSION 1245 04MAY1983 ULAPRW
C *****
C PRINT 1 LAYER ARRAY
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION BUF(NCOL,NROW),TEXT(4)
C -----
C
C1-----PRINT A HEADER
      IF(ILAY.LE.0) GO TO 5
      WRITE(IOUT,1) TEXT,ILAY,KSTP,KPER
      1 FORMAT(1H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
      1 ' IN STRESS PERIOD',I3/11X,71('-'))
C
C2-----MAKE SURE THE FORMAT CODE (IP OR IPRN) IS
C2-----BETWEEN 1 AND 12.
      5 IP=IPRN
      IF(IP.LT.1 .OR. IP.GT.12) IP=12
C
C3-----CALL THE UTILITY MODULE UCOLNO TO PRINT COLUMN NUMBERS.
      IF(IP.EQ.1) CALL UCOLNO(1,NCOL,0,11,11,IOUT)
      IF(IP.EQ.2) CALL UCOLNO(1,NCOL,0,9,14,IOUT)
      IF(IP.GT.2 .AND. IP.LT.7) CALL UCOLNO(1,NCOL,3,15,8,IOUT)
      IF(IP.GT.6 .AND. IP.LT.12) CALL UCOLNO(1,NCOL,3,20,6,IOUT)
      IF(IP.EQ.12) CALL UCOLNO(1,NCOL,0,10,12,IOUT)
C
C4-----LOOP THROUGH THE ROWS PRINTING EACH ONE IN ITS ENTIRETY.
      DO 1000 I=1,NROW
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120), IP
C
C----- FORMAT 11G10.3
      10 WRITE(IOUT,11) I,(BUF(J,I),J=1,NCOL)
      11 FORMAT(1H0,I3,2X,1PG10.3,10(1X,G10.3)/(5X,11(1X,G10.3)))
      GO TO 1000
C
C----- FORMAT 9G13.6
      20 WRITE(IOUT,21) I,(BUF(J,I),J=1,NCOL)
      21 FORMAT(1H0,I3,2X,1PG13.6,8(1X,G13.6)/(5X,9(1X,G13.6)))
      GO TO 1000
C
C----- FORMAT 15F7.1
      30 WRITE(IOUT,31) I,(BUF(J,I),J=1,NCOL)
      31 FORMAT(1H0,I3,1X,15(1X,F7.1)/(5X,15(1X,F7.1)))
      GO TO 1000
C
C----- FORMAT 15F7.2
      40 WRITE(IOUT,41) I,(BUF(J,I),J=1,NCOL)
      41 FORMAT(1H0,I3,1X,15(1X,F7.2)/(5X,15(1X,F7.2)))
      GO TO 1000

```

```

C
C----- FORMAT 15F7.3
  50 WRITE(IOUT,51) I,(BUF(J,I),J=1,NCOL)
  51 FORMAT(1H0,I3,1X,15(1X,F7.3)/(5X,15(1X,F7.3)))
      GO TO 1000
C
C----- FORMAT 15F7.4
  60 WRITE(IOUT,61) I,(BUF(J,I),J=1,NCOL)
  61 FORMAT(1H0,I3,1X,15(1X,F7.4)/(5X,15(1X,F7.4)))
      GO TO 1000
C
C----- FORMAT 20F5.0
  70 WRITE(IOUT,71) I,(BUF(J,I),J=1,NCOL)
  71 FORMAT(1H0,I3,1X,20(1X,F5.0)/(5X,20(1X,F5.0)))
      GO TO 1000
C
C----- FORMAT 20F5.1
  80 WRITE(IOUT,81) I,(BUF(J,I),J=1,NCOL)
  81 FORMAT(1H0,I3,1X,20(1X,F5.1)/(5X,20(1X,F5.1)))
      GO TO 1000
C
C----- FORMAT 20F5.2
  90 WRITE(IOUT,91) I,(BUF(J,I),J=1,NCOL)
  91 FORMAT(1H0,I3,1X,20(1X,F5.2)/(5X,20(1X,F5.2)))
      GO TO 1000
C
C----- FORMAT 20F5.3
 100 WRITE(IOUT,101) I,(BUF(J,I),J=1,NCOL)
 101 FORMAT(1H0,I3,1X,20(1X,F5.3)/(5X,20(1X,F5.3)))
      GO TO 1000
C
C----- FORMAT 20F5.4
 110 WRITE(IOUT,111) I,(BUF(J,I),J=1,NCOL)
 111 FORMAT(1H0,I3,1X,20(1X,F5.4)/(5X,20(1X,F5.4)))
      GO TO 1000
C
C----- FORMAT 10G11.4
 120 WRITE(IOUT,121) I,(BUF(J,I),J=1,NCOL)
 121 FORMAT(1H0,I3,2X,1PG11.4,9(1X,G11.4)/(5X,10(1X,G11.4)))
C
1000 CONTINUE
C
C5-----RETURN
      RETURN
      END

```

List of Variables for Module ULAPRW

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BUF	Module	Buffer containing data to be printed or recorded.
I	Module	Index for rows.
ILAY	Module	Layer number.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IP	Module	Format code.
IPRN	Module	Code for the format to be used when printing arrays.
J	Module	Index for columns.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NROW	Global	Number of rows in the grid.
TEXT	Module	Label to be printed or recorded with the array data.

### Narrative for Module UCOLNO

Module UCOLNO prints column numbers at the top of a page when arrays of numbers are printed. It performs its functions in the following order:

1. Calculate the number of columns to be printed (NLBL), the width of a line (NTOT), and the number of lines needed to print all of the column numbers (NWRAP). Initialize the fields J1 and J2 which contain the first and last column number on each print line.

2. Build and print each line (DO STEPS 3-6).

3. Clear the line buffer (BF) in which the line is built.

4. Determine the first (J1) and last (J2) column number for the current line.

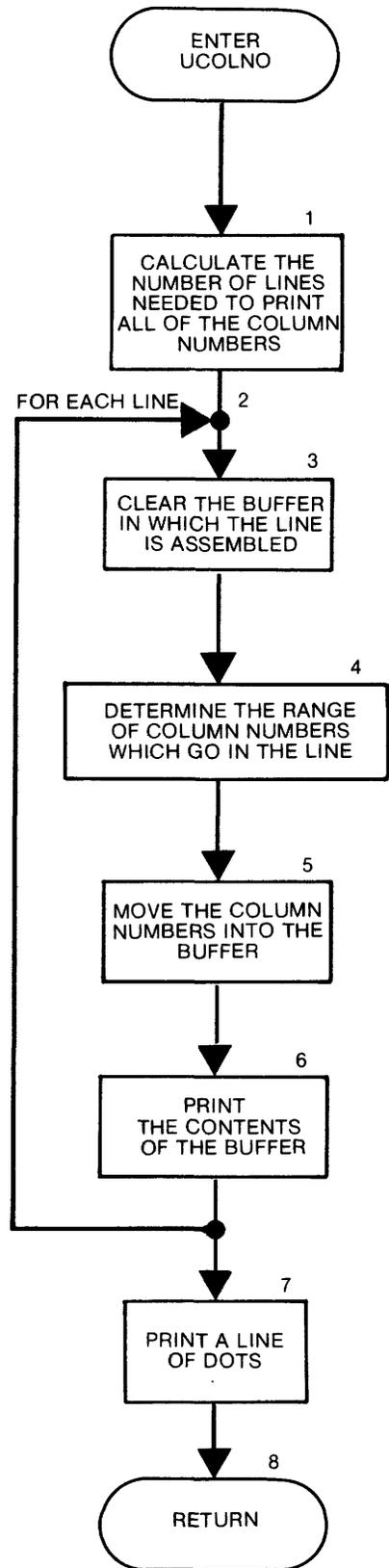
5. Put the column numbers in the line buffer. They are selected from the array DG. The indices I1, I2, and I3 point to the units digit, tens digit, and hundreds digit, respectively.

6. Print the line.

7. Print a line of dots.

8. RETURN.

Flow Chart for Module UCOLNO



```

SUBROUTINE UCOLNO(NLBL1,NLBL2,NSPACE,NCPL,NDIG,IOUT)
C
C
C-----VERSION 1446 20APR1983 UCOLNO
C *****
C OUTPUT COLUMN NUMBERS ABOVE A MATRIX PRINTOUT
C NLBL1 IS THE START COLUMN LABEL (NUMBER)
C NLBL2 IS THE STOP COLUMN LABEL (NUMBER)
C NSPACE IS NUMBER OF BLANK SPACES TO LEAVE AT START OF LINE
C NCPL IS NUMBER OF COLUMN NUMBERS PER LINE
C NDIG IS NUMBER OF CHARACTERS IN EACH COLUMN FIELD
C IOUT IS OUTPUT CHANNEL
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION BF(130),DG(10)
C
C DATA DG(1),DG(2),DG(3),DG(4),DG(5),DG(6),DG(7),DG(8),DG(9),DG(10)/
1 4H0 ,4H1 ,4H2 ,4H3 ,4H4 ,4H5 ,4H6 ,
2 4H7 ,4H8 ,4H9 /
C DATA DOT,SPACE/4H. ,4H /
C -----
C
C1-----CALCULATE # OF COLUMNS TO BE PRINTED (NLBL), WIDTH
C1-----OF A LINE (NTOT), NUMBER OF LINES (NWRAP).
WRITE(IOUT,1)
1 FORMAT(1X)
NLBL=NLBL2-NLBL1+1
N=NLBL
IF(NLBL.GT.NCPL) N=NCPL
NTOT=NSPACE+N*NDIG
IF(NTOT.GT.130) GO TO 50
NWRAP=(NLBL-1)/NCPL + 1
J1=NLBL1-NCPL
J2=NLBL1-1
C
C2-----BUILD AND PRINT EACH LINE
DO 40 N=1,NWRAP
C
C3-----CLEAR THE BUFFER (BF).
DO 20 I=1,130
BF(I)=SPACE
20 CONTINUE
NBF=NSPACE
C
C4-----DETERMINE FIRST (J1) AND LAST (J2) COLUMN # FOR THIS LINE.
J1=J1+NCPL
J2=J2+NCPL
IF(J2.GT.NLBL2) J2=NLBL2
C5-----LOAD THE COLUMN #'S INTO THE BUFFER.
DO 30 J=J1,J2
NBF=NBF+NDIG
I2=J/10
I1=J-I2*10+1
BF(NBF)=DG(I1)
IF(I2.EQ.0) GO TO 30
I3=I2/10
I2=I2-I3*10+1
BF(NBF-1)=DG(I2)
IF(I3.EQ.0) GO TO 30
BF(NBF-2)=DG(I3+1)
30 CONTINUE
C
C6-----PRINT THE CONTENTS OF THE BUFFER (I.E. PRINT THE LINE).
WRITE(IOUT,31) (BF(I),I=1,NBF)
31 FORMAT(1X,130A1)
C
40 CONTINUE
C
C7-----PRINT A LINE OF DOTS (FOR ESTHETIC PURPOSES ONLY).
50 NTOT=NTOT+5
IF(NTOT.GT.130) NTOT=130
WRITE(IOUT,51) (DOT,I=1,NTOT)
51 FORMAT(1X,130A1)
C
C8-----RETURN
RETURN
END

```

List of Variables for Module UCOLNO

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
BF	Module	DIMENSION (130), Buffer in which a line is assembled.
DG	Module	DIMENSION (10), Digits 0 through 9.
DOT	Module	Field containing a period.
I	Module	Index for BF.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
I1	Module	Index for DG (units digit).
I2	Module	Index for DG (tens digit).
I3	Module	Index for DG (hundreds digit).
J	Module	Index for column numbers.
J1	Module	First column number on the current line.
J2	Module	Last column number on the current line.
N	Module	Number of column numbers.
NBF	Module	Index for BF.
NCPL	Module	Number of column numbers per line.
NDIG	Module	Number of characters in each column number field.
NLBL	Module	Number of column numbers to be printed.
NLBL1	Module	Start column number.
NLBL2	Module	Stop column number.
NSPACE	Module	Number of blank spaces at start of line.
NTOT	Module	Total number of characters on a line.
NWRAP	Module	Number of lines needed in wrap format.
SPACE	Module	Field containing blanks.

## Narrative for Module U2DREL

Module U2DREL reads values for a two-dimensional real array. First it reads an "array-control record." Then, based on the contents of the array-control record, it may read array values. The array-control record contains four fields: location (LOCAT), constant (CNSTNT), format (FMTIN), and printout indicator (IPRN). The LOCAT field determines where array values will come from. If LOCAT is positive, it is the unit number from which array values will be read in the format specified in FMTIN. If LOCAT is negative, the sign is reversed to give the unit number from which an unformatted record containing the array values will be read. (Before the array record is read, a record will be read and ignored. Thus output from the module ULASAV can be read.) If LOCAT is zero, all of the array values will be set equal to CNSTNT. When LOCAT is not zero and CNSTNT is not zero, the array values will be multiplied by the value of CNSTNT. The field IPRN contains a code number for a FORTRAN format to be used when printing the array.

Module U2DREL performs its tasks in the following order:

1. Read the array-control record (LOCAT, CNSTNT, FMTIN, and IPRN).
2. Use LOCAT to determine where the array values are coming from.  
GO TO STEPS 3, 4, OR 5.
3. If LOCAT equals zero, set all array values equal to CNSTNT, print a message to that effect, and RETURN.
4. If LOCAT is greater than zero, read array values according to the format in FMTIN. GO TO STEP 6.
5. If LOCAT is less than zero, read an unformatted dummy record and then read an unformatted record containing the array values. GO TO STEP 6.
6. If CNSTNT is not equal to zero, multiply array values by CNSTNT.
7. If IPRN is greater than or equal to zero, call utility module ULAPRW to print the array values using IPRN as the format code.
8. RETURN.

## Flow Chart for Module U2DREL

Array Control Record controls the input of array values. It contains four fields: LOCAT, CNSTNT, FMTIN, and IPRN.

LOCAT is a code showing where array values will come from.

If  $LOCAT < 0$ , array values will be read from an unformatted record from a unit number equal to  $-LOCAT$ .

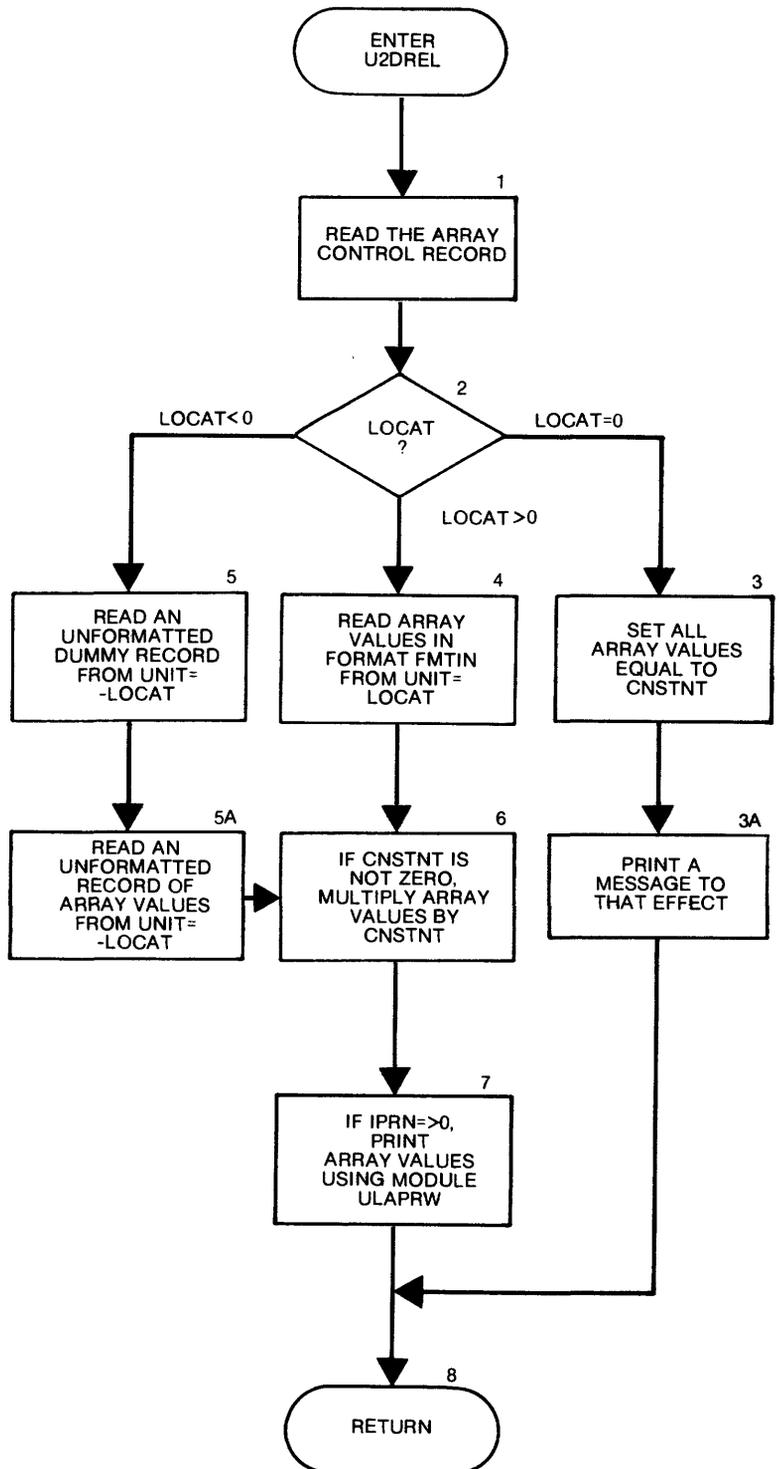
If  $LOCAT = 0$ , array values will be set equal to CNSTNT.

If  $LOCAT > 0$ , array values will be read from the unit number equal to LOCAT in the format specified in FMTIN.

CNSTNT is the constant to which all array values are set if LOCAT is equal to zero, and it is the constant by which all array values are multiplied if LOCAT is not equal to zero.

FMTIN is the format in which array values are read if LOCAT is greater than zero.

IPRN is a code showing the format to be used if array values are to be printed.



```

SUBROUTINE U2DREL(A, ANAME, II, JJ, K, IN, IOUT)
C
C
C-----VERSION 1439 20APR1983 U2DREL
C*****
C ROUTINE TO INPUT 2-D REAL DATA MATRICES
C   A IS ARRAY TO INPUT
C   ANAME IS 24 CHARACTER DESCRIPTION OF A
C   II IS NO. OF ROWS
C   JJ IS NO. OF COLS
C   K IS LAYER NO. (USED WITH NAME TO TITLE PRINTOUT UNLESS K IS 0)
C   IN IS INPUT UNIT
C   IOUT IS OUTPUT UNIT
C*****
C
C SPECIFICATIONS:
C-----
C DIMENSION A(JJ,II), ANAME(6), FMTIN(5)
C-----
C
C1-----READ ARRAY CONTROL RECORD.
   READ (IN,1) LOCAT, CNSTNT, FMTIN, IPRN
   1 FORMAT(I10, F10.0, 5A4, I10)
C
C2-----USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
   IF(LOCAT) 200, 50, 90
C
C3-----IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT. RETURN
   50 DO 80 I=1, II
      DO 80 J=1, JJ
      80 A(J,I)=CNSTNT
      IF(K.GT.0) WRITE(IOUT,2) ANAME, CNSTNT, K
      2 FORMAT(1H0, //30X, 6A4, ' = ', G15.7, ' FOR LAYER ', I3)
      IF(K.LE.0) WRITE(IOUT,3) ANAME, CNSTNT
      3 FORMAT(1H0, 52X, 6A4, ' = ', G15.7)
      RETURN
C
C4-----IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 IF(K.GT.0) WRITE(IOUT,4) ANAME, K, LOCAT, FMTIN
      4 FORMAT(1H0, //30X, 6A4, ' FOR LAYER ', I3, ' WILL BE READ ON UNIT ',
      1 I3, ' USING FORMAT: ', 5A4/30X, 96('-'))
      IF(K.LE.0) WRITE(IOUT,5) ANAME, LOCAT, FMTIN
      5 FORMAT(1H0, //30X, 6A4, ' WILL BE READ ON UNIT ',
      1 I3, ' USING FORMAT: ', 5A4/30X, 83('-'))
      DO 100 I=1, II
      READ (LOCAT, FMTIN) (A(J,I), J=1, JJ)
      100 CONTINUE
      GO TO 300
C
C5-----LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY VALUES
   200 LOCAT=-LOCAT
      IF(K.GT.0) WRITE(IOUT,201) ANAME, K, LOCAT
      201 FORMAT(1H0, //30X, 6A4, ' LAYER ', I3,
      1 ' WILL BE READ UNFORMATTED ON UNIT ', I3/30X, 73('-'))
      IF(K.LE.0) WRITE(IOUT,202) ANAME, LOCAT
      202 FORMAT(1H0, //30X,
      1 ' WILL BE READ UNFORMATTED ON UNIT ', I3/30X, 60('-'))
C
C5A-----READ AN UNFORMATTED DUMMY RECORD FIRST.
   READ(LOCAT)
   READ(LOCAT) A
C
C6-----IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
   300 IF(CNSTNT.EQ.0.) GO TO 320
      DO 310 I=1, II
      DO 310 J=1, JJ
      A(J,I)=A(J,I)*CNSTNT
      310 CONTINUE
C
C7-----IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
   320 IF(IPRN.LT.0) RETURN
      CALL ULAPRW(A, ANAME, 0, 0, JJ, II, 0, IPRN, IOUT)
      RETURN
C
C8-----RETURN
   END

```

List of Variables for Module U2DREL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
A	Module	DIMENSION (JJ,II), Input array.
ANAME	Module	Label for printout of the input array.
CNSTNT	Module	Constant to which all array values are set if LOCAT is equal to zero or by which all array values are multiplied if LOCAT is not equal to zero.
FMTIN	Module	DIMENSION (5), Format under which array values will be read.
I	Module	Index for rows.
II	Module	Number of rows.
IN	Module	Unit number from which the array control record will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRN	Module	Code for format to be used when printing the arrays.
J	Module	Index for columns.
JJ	Module	Number of columns.
K	Module	Layer number.
LOCAT	Module	Location of values to fill in the array. < 0, read an unformatted record containing the array values. = 0, set all the array values equal to constant (CNSTNT). > 0, read the formatted records containing the array values.

## Narrative for Module U2DINT

Module U2DINT reads values for a two-dimensional integer array. First it reads an "array-control record." Then, based on the contents of the array-control record, it may read array values. The array-control record contains four fields: location (LOCAT), constant (ICONST), format (FMTIN), and printout indicator (IPRN). The LOCAT field determines where array values will come from. If LOCAT is positive, it is the unit number from which array values will be read in the format specified in FMTIN. If LOCAT is negative, the sign is reversed to give the unit number from which an unformatted record containing the array values will be read. (Before the array record is read, a record will be read and ignored. Thus output from the module ULASAV can be read.) If LOCAT is zero, all of the array values will be set equal to ICONST. When LOCAT is not zero and ICONST is not zero, the array values will be multiplied by the value of ICONST. The field IPRN (table 1) contains a code number for a FORTRAN format to be used when printing the array.

Module U2DINT performs its tasks in the following order:

1. Read the array-control record (LOCAT, ICONST, FMTIN, and IPRN).
2. Use LOCAT to determine where the array is coming from. GO TO STEPS 3, 4, OR 5.
3. If LOCAT equals zero, set all array values equal to CNSTNT, print a message to that effect, and RETURN.
4. If LOCAT is greater than zero, read array values according to the format in FMTIN. GO TO STEP 6.
5. If LOCAT is less than zero, read an unformatted dummy record and then read an unformatted record containing the array values. GO TO STEP 6.
6. If ICONST is not equal to zero, multiply array values by ICONST.
7. If IPRN is greater than or equal to zero, print the array values using IPRN as the format code.
8. Call utility module UCOLNO to print column numbers at the top of the page.
9. Print each row in the array.
10. Select the format for printing.
11. RETURN.

## Flow Chart for Module U2DINT

Array Control Record controls the input of array values. It contains four fields: LOCAT, ICONST, FMTIN, and IPRN.

LOCAT is a code showing where array values will come from.

If  $LOCAT < 0$ , array values will be read from an unformatted record from a unit number equal to  $-LOCAT$ .

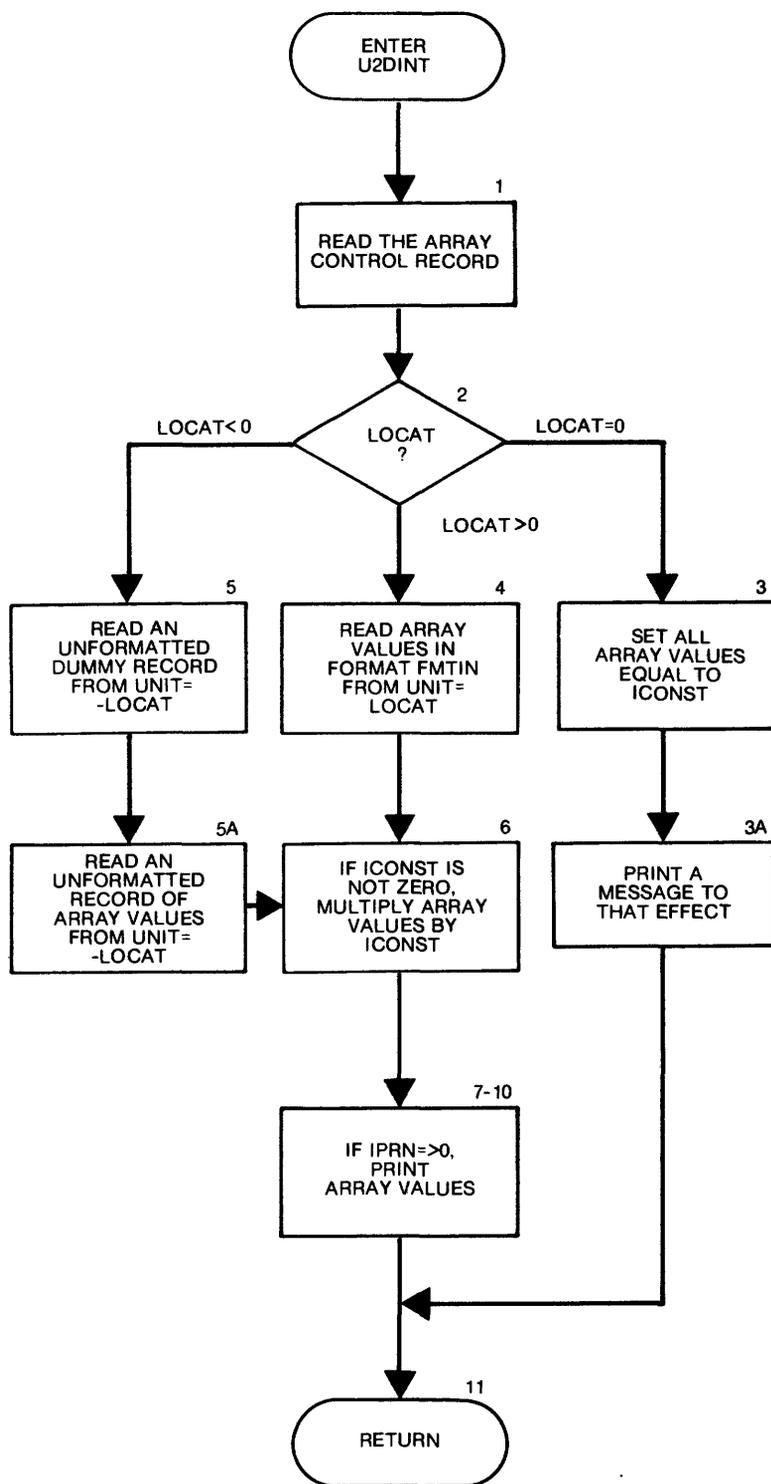
If  $LOCAT = 0$ , array values will be set equal to ICONST.

If  $LOCAT > 0$ , array values will be read from the unit number equal to LOCAT in the format specified in FMTIN.

ICONST is the constant to which all array values are set if LOCAT is equal to zero, and it is the constant by which all array values are multiplied if LOCAT is not equal to zero.

FMTIN is the format in which array values are read if LOCAT is greater than zero.

IPRN is a code showing the format to be used if array values are to be printed.



```

SUBROUTINE U2DINT(IA,ANAME,II,JJ,K,IN,IOUT)
C
C
C-----VERSION 1442 20APR1983 U2DINT
C *****
C ROUTINE TO INPUT 2-D INTEGER DATA MATRICES
C IA IS ARRAY TO INPUT
C ANAME IS 24 CHARACTER DESCRIPTION OF IA
C II IS NO. OF ROWS
C JJ IS NO. OF COLS
C K IS LAYER NO. (USED WITH NAME TO TITLE PRINTOUT UNLESS K IS 0)
C IN IS INPUT UNIT
C IOUT IS OUTPUT UNIT
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION IA(JJ,II),ANAME(6),FMTIN(5)
C -----
C
C1-----READ ARRAY CONTROL RECORD.
READ (IN,1) LOCAT,ICONST,FMTIN,IPRN
1 FORMAT(I10,I10,5A4,I10)
C
C2-----USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
IF(LOCAT) 200,50,90
C
C3-----IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO ICONST. RETURN
50 DO 80 I=1,II
DO 80 J=1,JJ
80 IA(J,I)=ICONST
IF(K.GT.0) WRITE(IOUT,2) ANAME,ICONST,K
2 FORMAT(1H0,52X,6A4,'=',I15,' FOR LAYER',I3)
IF(K.LE.0) WRITE(IOUT,3) ANAME,ICONST
3 FORMAT(1H0,52X,6A4,'=',I15)
RETURN
C
C4-----IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
90 IF(K.GT.0) WRITE(IOUT,4) ANAME,K,LOCAT,FMTIN
4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
1 I3,' USING FORMAT: ',5A4/30X,96('-'))
IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
1 I3,' USING FORMAT: ',5A4/30X,83('-'))
DO 100 I=1,II
READ (LOCAT,FMTIN) (IA(J,I),J=1,JJ)
100 CONTINUE
GO TO 300
C
C5-----LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY VALUES
200 LOCAT=-LOCAT
IF(K.GT.0) WRITE(IOUT,201) ANAME,K,LOCAT
201 FORMAT(1H0,///30X,6A4,' LAYER',I3,
1 ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,73('-'))
IF(K.LE.0) WRITE(IOUT,202) ANAME,LOCAT
202 FORMAT(1H0,///30X,6A4,
1 ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,60('-'))
C
C5A-----READ AN UNFORMATTED DUMMY RECORD FIRST.
READ(LOCAT)
READ(LOCAT) IA
C
C6-----IF ICONST NOT ZERO THEN MULTIPLY ARRAY VALUES BY ICONST.
300 IF(ICONST.EQ.0) GO TO 320
DO 310 I=1,II
DO 310 J=1,JJ
IA(J,I)=IA(J,I)*ICONST
310 CONTINUE
C
C7-----IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
320 IF(IPRN.LT.0) RETURN
IF(IPRN.GT.5) IPRN=0
IPRN=IPRN+1

```

```

C
C8-----PRINT COLUMN NUMBERS AT TOP OF PAGE.
      IF(IPRN.EQ.1) CALL UCOLNO(1,JJ,0,10,12,IOUT)
      NL=125/IPRN/5*5
      IF(IPRN.GT.1) CALL UCOLNO(1,JJ,4,NL,IPRN,IOUT)
C
C9-----PRINT EACH ROW IN THE ARRAY.
      DO 110 I=1,II
C
C10-----SELECT THE FORMAT
      GO TO(101,102,103,104,105,106), IPRN
C
C-----FORMAT 1011
      101 WRITE(IOUT,1001) I,(IA(J,I),J=1,JJ)
      1001 FORMAT(1H0,I3,2X,I11,9(1X,I11)/(5X,10(1X,I11)))
      GO TO 110
C
C-----FORMAT 6011
      102 WRITE(IOUT,1002) I,(IA(J,I),J=1,JJ)
      1002 FORMAT(1H0,I3,1X,60(1X,I1)/(5X,60(1X,I1)))
      GO TO 110
C
C-----FORMAT 4012
      103 WRITE(IOUT,1003) I,(IA(J,I),J=1,JJ)
      1003 FORMAT(1H0,I3,1X,40(1X,I2)/(5X,40(1X,I2)))
      GO TO 110
C
C-----FORMAT 3013
      104 WRITE(IOUT,1004) I,(IA(J,I),J=1,JJ)
      1004 FORMAT(1H0,I3,1X,30(1X,I3)/(5X,30(1X,I3)))
      GO TO 110
C
C-----FORMAT 2514
      105 WRITE(IOUT,1005) I,(IA(J,I),J=1,JJ)
      1005 FORMAT(1H0,I3,1X,25(1X,I4)/(5X,25(1X,I4)))
      GO TO 110
C
C-----FORMAT 2015
      106 WRITE(IOUT,1006) I,(IA(J,I),J=1,JJ)
      1006 FORMAT(1H0,I3,1X,20(1X,I5)/(5X,20(1X,I5)))
      110 CONTINUE
      RETURN
C
C11-----RETURN
      END

```

List of Variables for Module U2DINT

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
ANAME	Module	Label for the printout of input array.
FMTIN	Module	DIMENSION (5), Format under which the array values will be read.
I	Module	Index for rows.
IA	Module	DIMENSION (JJ,II), Input array.
ICONST	Module	Constant to which all array values are set if LOCAT is equal to zero or by which all array values are multiplied if LOCAT is not equal to zero.
II	Module	Number of rows.
IN	Module	Unit number from which the array-control record will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRN	Module	Code for format to be used when printing arrays.
J	Module	Index for columns.
JJ	Module	Number of columns.
K	Module	Layer number.
LOCAT	Module	Location of values to fill in the array. < 0, read an unformatted record containing the array values. = 0, set all the array values equal to constant (CNSTNT). > 0, read formatted records containing the array values.
NL	Module	Number of columns per line.

## Narrative for Module UIDREL

Module UIDREL reads values for a one-dimensional real array. First it reads an "array-control record." Then, based on the contents of the array-control record, it may read array values. The array-control record contains four fields: location (LOCAT), constant (CNSTNT), format (FMTIN), and printout indicator (IPRN). The LOCAT field determines where array values will come from. If LOCAT is positive, it is the unit number from which array values will be read in the format specified in FMTIN. If LOCAT is zero, all of the array values will be set equal to CNSTNT. If LOCAT is not zero and CNSTNT is not zero, the array values will be multiplied by the value of CNSTNT. The field IPRN (table 2) contains a code number for a FORTRAN format to be used when printing the array.

Module UIDREL performs its tasks in the following order:

1. Read the array-control record (LOCAT, CNSTNT, FMTIN, and IPRN).
2. Use LOCAT to determine where the array is coming from (DO STEPS 3 OR 4).
3. If LOCAT equals zero, set all array values equal to CNSTNT and print a message to that effect. RETURN.
4. If LOCAT is greater than zero, read array values according to the format in FMTIN.
5. If CNSTNT is not equal to zero, multiply the array values by CNSTNT.
6. If IPRN is greater than or equal to zero, print the array values.
7. RETURN.

## Flow Chart for Module U1DREL

Array Control Record controls the input of array values. It contains four fields: LOCAT, CNSTNT, FMTIN, and IPRN.

LOCAT is a code showing where array values will come from.

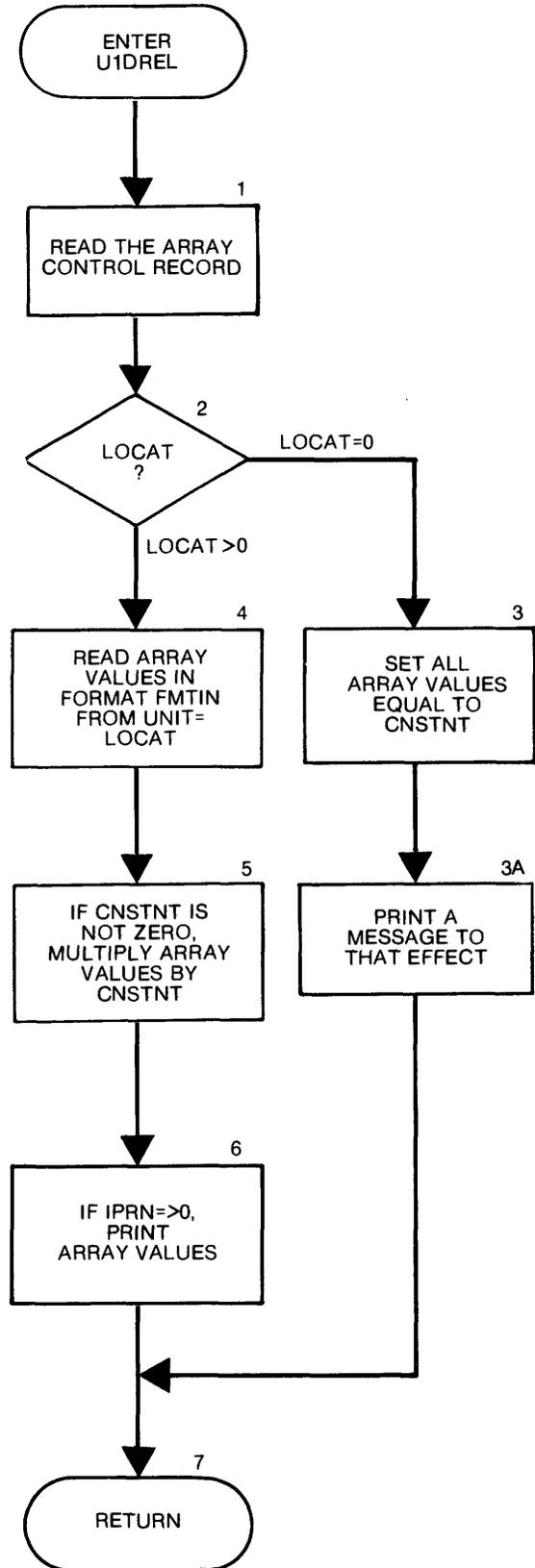
If LOCAT = 0, array values will be set equal to CNSTNT.

If LOCAT > 0, array values will be read from the unit number equal to LOCAT in the format specified in FMTIN.

CNSTNT is the constant to which all array values are set if LOCAT is equal to zero, and it is the constant by which all array values are multiplied if LOCAT is not equal to zero.

FMTIN is the format in which array values are read if LOCAT is greater than zero.

IPRN is a code showing the format to be used if array values are to be printed.



```

SUBROUTINE UIDREL(A, ANAME, JJ, IN, IOUT)
C
C
C-----VERSION 1436 20MAY1983 UIDREL
C *****
C ROUTINE TO INPUT 1-D REAL DATA MATRICES
C   A IS ARRAY TO INPUT
C   ANAME IS 24 CHARACTER DESCRIPTION OF A
C   JJ IS NO. OF ELEMENTS
C   IN IS INPUT UNIT
C   IOUT IS OUTPUT UNIT
C *****
C
C SPECIFICATIONS:
C -----
C DIMENSION A(JJ), ANAME(6), FMTIN(5)
C -----
C
C1-----READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT, CNSTNT, FMTIN, IPRN
      1 FORMAT(I10, F10.0, 5A4, I10)
C
C2-----USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT.GT.0) GO TO 90
C
C3-----IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT. RETURN
      DO 80 J=1, JJ
      80 A(J)=CNSTNT
      WRITE(IOUT,3) ANAME, CNSTNT
      3 FORMAT(1H0, 52X, 6A4, ' = ', G15.7)
      RETURN
C
C4-----IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
      90 WRITE(IOUT,5) ANAME, LOCAT, FMTIN
      5 FORMAT(1H0, //, 30X, 6A4, ' WILL BE READ ON UNIT', I3,
      1 ' USING FORMAT: ', 5A4/30X, 79('-', '/'))
      READ (LOCAT, FMTIN) (A(J), J=1, JJ)
C
C5-----IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
      IF(CNSTNT.EQ.0.) GO TO 120
      DO 100 J=1, JJ
      100 A(J)=A(J)*CNSTNT
C
C6-----IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
      120 IF(IPRN.LT.0) RETURN
      WRITE(IOUT,1001) (A(J), J=1, JJ)
      1001 FORMAT((1X, 1PG12.5, 9(1X, G12.5)))
      RETURN
C
C7-----CONTINUE
      END

```

List of Variables for Module UIDREL

<u>Variable</u>	<u>Range</u>	<u>Definition</u>
A	Module	DIMENSION (JJ), Input array.
ANAME	Module	Label for printout of the input array.
CNSTNT	Module	Constant to which all array values are set if LOCAT is equal to zero or by which all array values are multiplied if LOCAT is not equal to zero.
FMTIN	Module	DIMENSION (5), Format under which the array values will be read.
IN	Module	Unit number from which the array control record will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRN	Module	Code for the format to be used when printing the arrays.
J	Module	Array index.
JJ	Module	Number of elements in the array.
LOCAT	Module	Location of values to fill in the array. < 0, read an unformatted record containing the array values. = 0, set all the array values equal to constant (CNSTNT). > 0, read the formatted records containing the array values.

## REFERENCES

- Trescott, Peter C., 1975, Documentation of finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Open-File Report 75-438, 32 p.
- Trescott, P. C., Pinder, G. F., and Larson, S. P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, Chapter C1, 116 p.
- Weinstein, H. C., Stone, H. L., and Kwan, T. V., 1969, Iterative procedure for solution of systems of parabolic and elliptic equations in three dimensions: Indus. Engineering Chemistry Fundamentals, v. 8, no. 2, p. 281-287.

## APPENDIX A

### PROGRAM PORTABILITY

#### Introduction

One of the major design requirements for the model program was that it should run with a minimum of modification on most computers that are physically capable of running a program of this type. A program that can be easily interchanged among computers is called portable. The program documented in this report is portable. It has run successfully without modification on computers manufactured by many different companies including IBM, Control Data, Digital Equipment Corporation, Cray, Prime, Amdahl, and Univac. The compilers which were used were extensions of either FORTRAN 66 or FORTRAN 77. The following discussion explains, in more detail, the concept of portability and the significance of double precision arithmetic.

#### The Impact of the Programming Language on Portability

The programming language is the most important factor that determines program portability. There are a variety of programming languages available, and, for each language, there are numerous versions which have resulted from the desire of vendors to improve the power of the language and to take advantage of the hardware features of their particular computers. The most commonly available language suitable to use for the model program is FORTRAN. There are two versions defined by the American National Standards Institute (ANSI) on which most commercial versions are based, ANSI X3.9-1966 and ANSI X3.9-1978 (ANSI, 1966 and ANSI, 1978). These versions are commonly referred to as FORTRAN 66 and FORTRAN 77, respectively. FORTRAN 66 was selected for the model. The only exception to the FORTRAN 66 standard is that character strings in FORMAT and DATA statements are in single quotes rather than using Hollerith constants. Most FORTRAN 66 compilers include this capability. Although FORTRAN 66 is the less powerful of the two, it is far more widely supported in its entirety than is FORTRAN 77. The additional features of FORTRAN 77 would have made programming more convenient, but these features were not necessary to make a computationally efficient program. It is recognized, though, that some users may want to use the program with FORTRAN 77. Only a few changes are required to convert the program to FORTRAN 77 because FORTRAN 66 is nearly a subset of FORTRAN 77. Information about how to convert is provided in a following section (see Conversion to FORTRAN 77). Most commercial versions of FORTRAN 66 include some extended features not defined as part of FORTRAN 66, but such features were not used because they are not widely available. Since the program adheres to the FORTRAN 66 standard as understood by the authors, the program should work on any computer supporting this language provided that the computer has adequate computational power.

#### The Impact of Computational Precision on Portability

Variation of precision among computers causes some problems with program portability. Computational precision refers to the accuracy with which numbers are calculated and stored in the computer. To prevent the

imposition of constraints on the computers that implement FORTRAN 66, the computational precision was not defined as part of the standard. The accuracy of model results is highly dependent on computational precision, so precision must be considered when moving the model program among computers. Also, numeric precision has an effect on alphanumeric data since alphanumeric characters are stored in numeric variables and arrays in FORTRAN 66.

The model program was developed on computers using 32 binary bits to represent single-precision real numbers. This gives about seven digits of precision and includes the range from  $10^{-39}$  to  $10^{38}$ . Double-precision real numbers are represented by 64 binary bits and have the same range as single-precision numbers, but about 16 decimal digits of precision. The head array, HNEW, and some variables in the solvers are stored as double precision, and, accordingly, some calculations are double precision. This was necessary for accuracy under some conditions. If using a computer with greater than 32-bit precision, the use of double-precision data and calculations may be unnecessary. The program should still work without modification, but computation time and memory would be saved if double precision were eliminated. One test of the program, on a computer using 64 bits to represent a real number, realized a savings of 75 percent of execution time when single-precision arithmetic was used exclusively. If it is determined that single precision is adequate, make the following changes to the program:

1. Delete all DOUBLE-PRECISION specification statements.

2. Change the Basic Package allocate module to allocate only NLAY\*NROW\*NCOL storage locations for HNEW rather than 2\*NLAY\*NROW\*NCOL locations.

There are no other statements that require change. This is because the use of double-precision functions and constants are avoided; all conversions between real and double precision are done by implied type changes in assignment statements, and there are no "D" formats. The "D" formats are avoided by moving HNEW to a single-precision buffer before printing.

If less than 32-bit precision for single-precision real numbers is available, double precision may be necessary for all calculations. To convert all real calculations to double precision, do the following:

1. Change all REAL arrays and variables to double precision by the special command available on some compilers or by declaring all REAL arrays and variables as DOUBLE PRECISION in the MAIN Program and in every subroutine. Any variable or array starting with the letters A-H or O-Z should be declared DOUBLE PRECISION.

2. Change the Basic Package allocate module to allocate only NLAY\*NROW\*NCOL storage locations for HNEW rather than 2\*NLAY\*NROW\*NCOL.

Some alphanumeric data (text data and format specifications) are stored in real variables and arrays. It is assumed that at least four characters can be stored in each variable or array element. It does not matter if more than four characters can fit because the extra characters are ignored.

Alphanumeric data occupy a trivial amount of memory, so the amount wasted if more than four characters can fit in a variable or array element is insignificant. If less than four characters can fit into a variable or array element, the program must be changed. It is unlikely, however, that less than four characters will fit into one variable or array element unless the computer precision is low enough that all REAL variables and arrays require conversion to double precision as described above. Such conversion would automatically solve the alphanumeric character problem. If, for some reason, less than four characters will fit into a REAL variable or array element and yet all REAL variables and arrays are not being converted to double precision, just the REAL variables and arrays that store alphanumeric data should be declared DOUBLE PRECISION.

### Conversion to FORTRAN 77

FORTRAN 66 is nearly a subset of FORTRAN 77. Accordingly, few changes are required to make the model program work on a FORTRAN 77 compiler. Most FORTRAN 77 compilers will accept the program as it is without change. Before modifying the program, try to compile it using your FORTRAN 77 compiler. Any changes that might have to be made are a result of the character data type which was added to FORTRAN 77. Any variables or arrays holding alphanumeric character data must be declared CHARACTER. This is most easily done by specifying that the same number of alphanumeric characters be stored in each character array element as was stored in each REAL variable or array element, which is four characters. This prevents the need to change the A4 format specifiers used to print alphanumeric characters. Thus, each DIMENSION ARRAY(N) statement, where ARRAY is an array holding 4\*N alphanumeric characters, should be changed to CHARACTER\*4 ARRAY(N). New statements of the form CHARACTER\*4 VAR, where VAR is a variable that holds alphanumeric data, should be added to every subroutine in which alphanumeric data are stored in a variable. Also, Hollerith constants in DATA statements must be changed to character constants. Note that Hollerith constants in FORMAT statements need not be changed. These constants are no longer called Hollerith, but rather H edit descriptors; however, the form and function are the same. Before making any of these changes, check the FORTRAN 77 manual for the particular computer being used to see if these changes are actually necessary. Some FORTRAN 77 compilers have extensions that allow FORTRAN 66 programs to run without modification.

The above mentioned modifications to convert the model program to FORTRAN 77 do not take advantage of the added features of FORTRAN 77. The modifications simply allow the program to work on a FORTRAN 77 compiler. Modifying the program to take advantage of new features probably would not be cost effective. Program execution efficiency would not be significantly improved. The only improvement would be in the area of program structure. For example, the number of program statements might be reduced by using the new structured programming statements of FORTRAN 77, and the sometimes confusing "GO TO" statements could be eliminated. However, because the model program is broken into short modules with simple logic, FORTRAN 77 structural statements would provide only small benefits. Also, changes to take advantage of FORTRAN 77 features would limit program portability until such time that FORTRAN 77 is widely supported.

## APPENDIX B

### SPACE REQUIREMENTS IN THE X ARRAY

The outline below gives the X-array space requirements for each package. The formulas can be used to calculate the exact size of the X array for a given problem. However, it is generally easier to simply run the model program and let it calculate the size. The total space required is printed as part of the model printout even if the X array is dimensioned too small. As a rough estimate, the X array is approximately 10 to 15 times the number of nodes in the model depending on the options selected. In the outline below, NODES is defined as  $NCOL * NROW * NLAY$ , the number of nodes in the model.

#### I. BAS Package

A.  $8 * NODES + (NLAY - 1) * NCOL * NROW + NROW + NCOL + 4 * NLAY$

B. Additionally, add NODES if start head is saved (ISTRN is not 0), and add NODES if BUFF is separate from RHS (IAPART is not 0)

#### II. BCF Package

A.  $NLAY$

B. If a transient simulation (ISS is 0), add NODES

C. For each layer where LAYCON is one or three, add  $2 * NCOL * NROW$

D. For each layer where LAYCON is two or three, add  $NCOL * NROW$

E. If a transient simulation (ISS is 0), for each layer where LAYCON is two or three, add  $NCOL * NROW$

#### III. WEL Package -- $4 * MXWELL$

IV. DRN Package --  $5 \cdot \text{MXDRAN}$

V. RIV Package --  $6 \cdot \text{MXRIVR}$

VI. EVT Package

A. Option 1 --  $3 \cdot \text{NCOL} \cdot \text{NROW}$

B. Option 2 --  $4 \cdot \text{NCOL} \cdot \text{NROW}$

VII. GHB Package --  $5 \cdot \text{MXBND}$

VIII. RCH Package

A. Options 1 and 3 --  $\text{NCOL} \cdot \text{NROW}$

B. Option 2 --  $2 \cdot \text{NCOL} \cdot \text{NROW}$

IX. SIP Package --  $4 \cdot \text{NODES} + 4 \cdot \text{MXITER} + \text{NPARM}$

X. SOR Package --  $(\text{NLAY} + 4) \cdot \text{NCOL} \cdot \text{NLAY} + 4 \cdot \text{MXITER}$

Generally, it is advisable to have the X-array dimension relatively close to the amount of space needed for a specific problem. On the other hand, it is inconvenient to redimension the X array every time a new option is selected. Several load modules with the size of the X array differing by a factor of two should be adequate.

## APPENDIX C

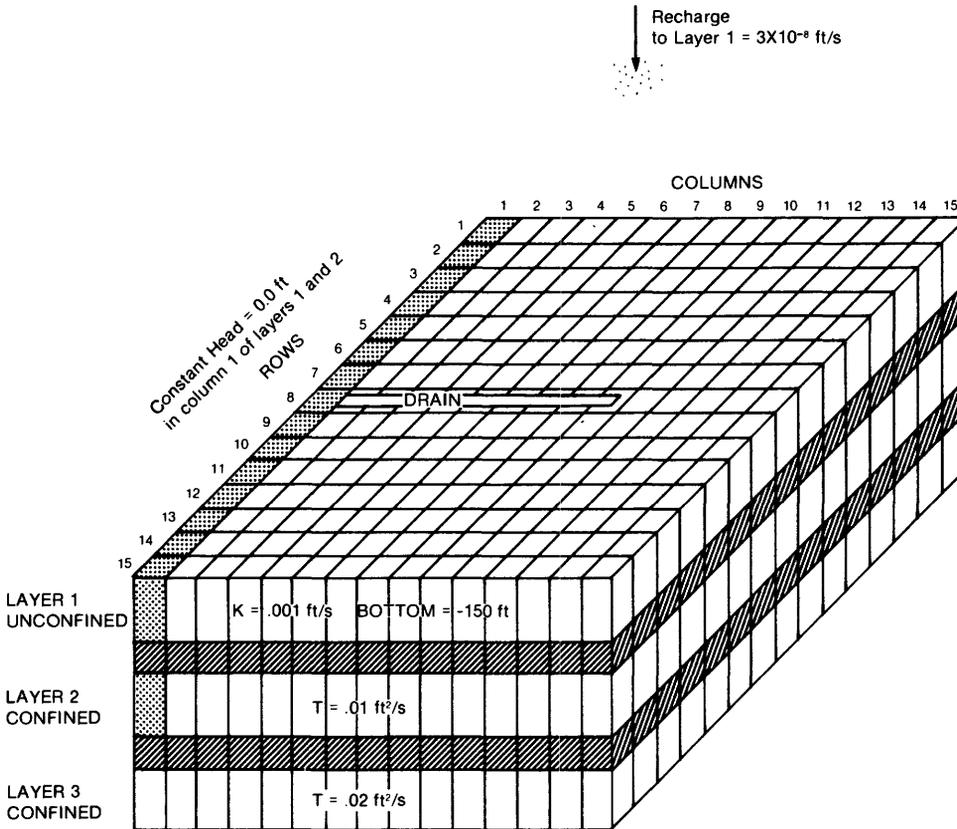
### CONTINUATION OF A PREVIOUS RUN

There is often value in breaking long simulations into several short model runs. This allows one to decide, between runs, whether or not to continue the simulation. Although the model program in this report does not have a special option for making continuation runs, it is quite simple to continue a simulation by using the output of one run as input of the next. Simply save the heads from the run that is to be continued on a disk file, and specify that file as starting heads for the next run. The subroutine that reads starting heads (U2DREL) is capable of reading model-generated disk files of saved heads without the need for reformatting. Because volumetric budget terms are always set to zero at the start of a model run, the printed budget on a model run represents only that one run, not the total of all runs in a series of continuation runs. If a total budget for a series of continuation runs is desired, the totals from each run can be added externally. Similarly, the model program keeps track of simulation time only for single model runs, but total simulation time for a series of continuation runs can be calculated externally by adding the simulation times of each run.

## APPENDIX D

### SAMPLE PROBLEM

This sample problem is intended to illustrate input and output from the program. There are three simulated layers, as shown in the accompanying illustration, which are separated from each other by confining layers. Each layer is a square 75,000 feet on a side and is divided by a grid into 15 rows and 15 columns which form squares 5,000 feet on a side. Flow within the confining layers is not simulated, but the effects of the confining layers on flow between the active layers are incorporated in the vertical conductivity (VCONT) terms. Flow into the system is infiltration from precipitation; flow out of the system is to buried drain tubes, discharging wells, and a lake which is represented by a constant-head boundary.



Between layers 1 and 2 vertical hydraulic conductivity divided by thickness =  $2 \times 10^{-8}/s$

Between layers 2 and 3 vertical hydraulic conductivity divided by thickness =  $1 \times 10^{-8}/s$

Setting starting heads equal to 0.0, the program was run to get a steady-state solution. The Strongly Implicit Procedure was used to solve the system of difference equations: the error criterion was set at 0.001 feet, the acceleration parameter was set to 1.0, and the maximum number of iterations was set equal to 50. A seed of 0.001 was specified for use in calculating the iteration parameters; 31 iterations were needed to close.

List of Wells

List of Drains

Q = 5 ft<sup>3</sup>/s for each well

Conductance = 1 ft<sup>2</sup>/s

<u>Layer</u>	<u>Row</u>	<u>Column</u>	<u>Layer</u>	<u>Row</u>	<u>Column</u>	<u>Elevation</u>
3	5	11	1	8	2	0.0
2	4	6	1	8	3	0.0
2	6	12	1	8	4	10.0
1	9	8	1	8	5	20.0
1	9	10	1	8	6	30.0
1	9	12	1	8	7	50.0
1	9	14	1	8	8	70.0
1	11	8	1	8	9	90.0
1	11	10	1	8	10	100.0
1	11	12				
1	11	14				
1	13	8				
1	13	10				
1	13	12				
1	13	14				



```

//FT11F001 DD *
  1      0      ISS,IBCABD
  1 0 0
  0      1.      TRPY
  0      5000.   DELR
  0      5000.   DELC
  0      .001    HY-1
  0      -150.   BOT-1
  0      2.E-8   VHY/THICK-1
  0      .01     T-2
  0      1.E-8   VHY/THICK-2
  0      .02     T-3

/*
//FT18F001 DD *
  1      0      NRCHOP,IRCHBD
  1      INRECH
  0      3.E-8   RECH-1

/*
//FT19F001 DD *
  50      5      MXITER,NPARM
  1.      .001   0      .001   1 ACCL,ERR,IPCALC,WSEE

//FT13F001 DD *
  9      0      MXDRAI,IDRNB
  9      NDRAIN
  1      8      2      0.      1.E00
  1      8      3      0.      1.E00
  1      8      4      10.     1.E00
  1      8      5      20.     1.E00
  1      8      6      30.     1.E00
  1      8      7      50.     1.E00
  1      8      8      70.     1.E00
  1      8      9      90.     1.E00
  1      8      10     100.    1.E00

//FT12F001 DD *
  15      0      MXWELL,IWELBD
  15      NWELL
  3      5      11      -5.
  2      4      6      -5.
  2      6      12     -5.
  1      9      8      -5.
  1      9      10     -5.
  1      9      12     -5.
  1      9      14     -5.
  1      11     8      -5.
  1      11     10     -5.
  1      11     12     -5.
  1      11     14     -5.
  1      13     8      -5.
  1      13     10     -5.
  1      13     12     -5.
  1      13     14     -5.

```

U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER MODEL  
SAMPLE--3 LAYERS,15 ROWS,15 COLUMNS,STEADY STATE,CONSTANT HEADS COLUMN 1,LAYERS 1 AND 2,RECHARGE,WELLS AND DRAINS  
3 LAYERS  
15 ROWS  
15 COLUMNS

1 STRESS PERIOD(S) IN SIMULATION

MODEL TIME UNIT IS SECONDS

I/O UNITS:

ELEMENT OF IUNIT: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24  
I/O UNIT: 11 12 13 0 0 0 0 18 19 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 12/08/83 INPUT READ FROM UNIT 1

ARRAYS RHS AND BUFF WILL SHARE MEMORY.

START HEAD WILL NOT BE SAVED -- DRAWDOWN CANNOT BE CALCULATED

5892 ELEMENTS IN X ARRAY ARE USED BY BAS

5892 ELEMENTS OF X ARRAY USED OUT OF 100000

BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1, 12/08/83 INPUT READ FROM UNIT 11

STEADY-STATE SIMULATION

LAYER AQUIFER TYPE

-----

1 1  
2 0  
3 0

453 ELEMENTS IN X ARRAY ARE USED BY BCF

6345 ELEMENTS OF X ARRAY USED OUT OF 100000

WELL1 -- WELL PACKAGE, VERSION 1, 12/08/83 INPUT READ FROM 12

MAXIMUM OF 15 WELLS

60 ELEMENTS IN X ARRAY ARE USED FOR WELLS

6405 ELEMENTS OF X ARRAY USED OUT OF 100000

DRN1 -- DRAIN PACKAGE, VERSION 1, 12/08/83 INPUT READ FROM UNIT 13

MAXIMUM OF 9 DRAINS

45 ELEMENTS IN X ARRAY ARE USED FOR DRAINS

6450 ELEMENTS OF X ARRAY USED OUT OF 100000

RCH1 -- RECHARGE PACKAGE, VERSION 1, 12/08/83 INPUT READ FROM UNIT 18

OPTION 1 -- RECHARGE TO TOP LAYER

225 ELEMENTS OF X ARRAY USED FOR RECHARGE

6675 ELEMENTS OF X ARRAY USED OUT OF 100000

SIP1 -- STRONGLY IMPLICIT PROCEDURE SOLUTION PACKAGE, VERSION 1, 12/08/83 INPUT READ FROM UNIT 19

MAXIMUM OF 50 ITERATIONS ALLOWED FOR CLOSURE

5 ITERATION PARAMETERS

2905 ELEMENTS IN X ARRAY ARE USED BY SIP

9580 ELEMENTS OF X ARRAY USED OUT OF 100000



AQUIFER HEAD WILL BE SET TO 999.99 AT ALL NO-FLOW NODES (IBOUND=0).  
 BOUNDARY ARRAY = 1 FOR LAYER 3  
 INITIAL HEAD = .0 FOR LAYER 1  
 INITIAL HEAD = .0 FOR LAYER 2  
 INITIAL HEAD = .0 FOR LAYER 3

DEFAULT OUTPUT CONTROL -- THE FOLLOWING OUTPUT COMES AT THE END OF EACH STRESS PERIOD:  
 TOTAL VOLUMETRIC BUDGET  
 HEAD

COLUMN TO ROW ANISOTROPY = 1.000000  
 DELR = 5000.000  
 DELC = 5000.000  
 HYD. COND. ALONG ROWS = .9999999E-03 FOR LAYER 1  
 BOTTOM = -150.0000 FOR LAYER 1  
 VERT HYD COND /THICKNESS = .2000000E-07 FOR LAYER 1  
 TRANSMIS. ALONG ROWS = .1000000E-01 FOR LAYER 2  
 VERT HYD COND /THICKNESS = .1000000E-07 FOR LAYER 2  
 TRANSMIS. ALONG ROWS = .2000000E-01 FOR LAYER 3

SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE

-----  
 MAXIMUM ITERATIONS ALLOWED FOR CLOSURE = 50  
 ACCELERATION PARAMETER = 1.0000  
 HEAD CHANGE CRITERION FOR CLOSURE = 0.10000E-02  
 SIP HEAD CHANGE PRINTOUT INTERVAL = 1

5 ITERATION PARAMETERS CALCULATED FROM SPECIFIED WSEED = 0.00100000 :

0.0 0.8221720E+00 0.9683772E+00 0.9943766E+00 0.9990000E+00  
 STRESS PERIOD NO. 1, LENGTH = 86400.00

-----  
 NUMBER OF TIME STEPS = 1  
 MULTIPLIER FOR DELT = 1.000  
 INITIAL TIME STEP SIZE = 86400.00

15 WELLS

LAYER	ROW	COL	STRESS RATE	WELL NO.
3	5	11	-5.0000	1
2	4	6	-5.0000	2
2	6	12	-5.0000	3
1	9	8	-5.0000	4
1	9	10	-5.0000	5
1	9	12	-5.0000	6
1	9	14	-5.0000	7
1	11	8	-5.0000	8
1	11	10	-5.0000	9
1	11	12	-5.0000	10
1	11	14	-5.0000	11
1	13	8	-5.0000	12
1	13	10	-5.0000	13
1	13	12	-5.0000	14
1	13	14	-5.0000	15

9 DRAINS

LAYER	ROW	COL	ELEVATION	CONDUCTANCE	DRAIN NO.
1	8	2	.0	1.000	1
1	8	3	.0	1.000	2
1	8	4	10.00	1.000	3
1	8	5	20.00	1.000	4
1	8	6	30.00	1.000	5
1	8	7	50.00	1.000	6
1	8	8	70.00	1.000	7
1	8	9	90.00	1.000	8
1	8	10	100.0	1.000	9

RECHARGE = .3000000E-07

31 ITERATIONS FOR TIME STEP 1 IN STRESS PERIOD 1

MAXIMUM HEAD CHANGE FOR EACH ITERATION:

| HEAD CHANGE LAYER, ROW, COL |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| -22.41 ( 3, 5, 11)          | 12.48 ( 1, 1, 15)           | 13.39 ( 3, 1, 14)           | 48.21 ( 1, 1, 15)           |
| 2.482 ( 1, 9, 14)           | 1.430 ( 3, 10, 13)          | 6.214 ( 1, 12, 14)          | 7.411 ( 3, 11, 14)          |
| .5503 ( 3, 8, 7)            | .4821 ( 2, 6, 9)            | .4711 ( 3, 5, 10)           | 2.019 ( 1, 11, 14)          |
| .1108 ( 1, 13, 12)          | .7059E-01 ( 3, 12, 11)      | .2819 ( 1, 14, 14)          | .3141 ( 3, 13, 14)          |
| .7855E-02 ( 1, 13, 12)      | .1586E-01 ( 2, 11, 11)      | .1777E-01 ( 3, 11, 10)      | .7911E-01 ( 1, 14, 14)      |
| .4169E-02 ( 1, 13, 14)      | .2555E-02 ( 3, 14, 15)      | .9771E-02 ( 1, 14, 14)      | .1082E-01 ( 3, 13, 14)      |
| .2426E-03 ( 1, 13, 12)      |                             |                             | .1030E-01 ( 1, 15, 15)      |
|                             |                             |                             | .8500E-01 ( 3, 7, 14)       |
|                             |                             |                             | 2.302 ( 3, 5, 13)           |
|                             |                             |                             | 13.66 ( 1, 15, 15)          |
|                             |                             |                             | 35.90 ( 3, 1, 13)           |

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

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15					
1	.0	24.94	44.01	59.26	71.82	82.52	91.91	100.0	106.9	112.6
2	117.4	121.3	124.3	126.4	127.4	80.57	90.12	98.40	105.3	111.0
3	.0	24.45	43.10	57.98	70.17	76.21	86.51	95.20	102.2	107.6
4	115.7	119.6	122.7	124.9	126.1	68.03	81.34	90.75	97.64	102.5
5	.0	23.45	41.30	55.43	66.78	66.74	77.09	85.76	92.22	96.15
6	112.0	116.1	119.6	122.1	123.4	61.21	71.19	79.85	86.47	90.82
7	.0	21.92	38.61	51.75	61.79	51.84	63.08	72.68	79.95	84.92
8	106.1	110.7	114.9	117.9	119.4	36.97	52.59	64.31	72.52	77.25
9	.0	19.73	34.92	47.32	57.69	45.27	52.95	55.38	65.15	66.07
10	97.29	103.1	108.8	112.5	114.3	50.11	54.93	57.55	62.95	65.55
11	.0	16.51	29.50	40.90	51.30	53.24	55.81	53.33	60.27	59.29
12	93.03	94.23	102.1	106.4	108.4	55.92	58.33	58.47	61.93	63.18
13	.0	11.55	21.10	31.21	41.40	58.04	59.91	56.75	62.59	60.91
14	88.60	91.66	96.43	99.82	101.8	60.08	63.17	64.52	67.25	68.79
15	.0	3.483	6.832	16.25	26.30	61.26	65.02	67.52	69.94	72.01
16	81.99	85.00	89.27	91.72	94.33					
17	.0	10.54	19.11	28.12	36.92					
18	73.93	73.79	80.84	80.17	86.49					
19	.0	14.62	25.86	35.38	43.49					
20	70.39	72.44	76.72	78.26	81.79					
21	.0	17.11	29.96	40.01	47.78					
22	66.43	65.45	72.22	71.04	77.62					
23	.0	18.68	32.56	43.07	50.81					
24	67.12	68.50	72.29	73.46	76.85					
25	.0	19.67	34.24	45.14	53.01					
26	67.22	65.75	71.90	70.35	76.48					
27	.0	20.27	35.27	46.48	54.61					
28	71.64	73.18	75.84	77.03	79.09					
29	.0	20.56	35.78	47.16	55.48					
30	74.29	76.22	78.22	79.66	80.82					

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

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15					
1	.0	24.66	43.73	59.02	71.61	82.32	91.72	99.86	106.7	112.5
2	117.2	121.1	124.1	126.2	127.3	80.36	89.93	98.22	105.1	110.8
3	.0	115.5	119.4	124.8	125.9	75.77	86.29	95.02	102.0	107.4
4	111.8	116.0	119.5	121.9	123.2	60.17	80.90	90.55	97.45	102.3
5	.0	105.4	110.4	114.8	117.7	66.30	76.85	85.57	92.00	95.41
6	91.09	102.1	108.6	112.4	114.2	60.98	70.98	79.65	86.28	90.54
7	.0	92.06	101.7	106.2	108.3	51.70	62.90	72.48	79.76	84.73
8	88.35	91.24	96.22	99.65	101.6	38.25	52.94	64.19	72.34	77.12
9	.0	81.81	84.86	89.10	91.59	45.16	52.86	56.13	65.08	66.79
10	73.87	74.48	80.77	80.84	86.38	49.91	54.76	57.48	62.79	65.49
11	.0	70.24	76.57	78.20	81.64	53.05	55.68	54.09	60.20	60.04
12	66.37	66.18	72.16	71.75	77.51	55.73	58.16	58.41	61.78	63.12
13	.0	66.98	72.15	73.40	76.69	57.85	59.78	57.50	62.53	61.65
14	67.16	66.48	71.84	71.06	76.37	59.88	62.99	64.39	67.08	68.66
15	.0	71.48	75.68	76.91	78.93	61.07	64.84	67.34	69.76	71.84
	74.11	76.04	78.04	79.49	80.65					

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

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15					
1	1.800	24.34	43.36	58.70	71.33	82.06	91.48	99.63	106.5	112.3
2	1.17.0	120.9	123.9	126.0	127.1	80.07	89.68	97.99	104.9	110.6
3	1.764	23.85	42.46	57.42	69.66	75.28	85.98	94.77	101.7	107.2
4	1.15.3	119.2	122.4	124.6	125.7	62.69	80.41	90.28	97.19	101.9
5	1.691	22.86	40.67	54.87	66.20	65.80	76.54	85.30	91.67	94.17
6	1.11.5	115.7	119.3	121.7	123.0	60.67	70.70	79.38	86.01	90.12
7	1.578	21.35	37.98	51.17	60.85	51.55	62.67	72.22	79.50	84.46
8	1.04.1	110.0	114.5	117.5	119.0	39.84	53.40	64.07	72.11	76.95
9	1.415	19.18	34.30	46.75	57.10	45.06	52.78	57.03	65.02	67.64
10	77.46	100.7	108.2	112.1	114.0	49.65	54.54	57.44	62.61	65.44
11	1.176	15.99	28.91	40.33	50.76	52.79	55.53	55.01	60.16	60.94
12	90.60	88.55	101.2	106.0	108.0	55.47	57.94	58.37	61.60	63.08
13	.8273	11.21	20.79	30.88	41.09	57.60	59.63	58.39	62.48	62.54
14	87.98	90.77	95.94	99.41	101.4	59.63	62.76	64.24	66.87	68.52
15	.4331	5.131	10.19	19.27	29.19	60.81	64.59	67.11	69.52	71.61
8	81.58	84.68	88.88	91.44	93.95	60.81	64.59	67.11	69.52	71.61
9	.7543	10.22	18.82	27.84	36.66	60.81	64.59	67.11	69.52	71.61
10	73.81	75.31	80.72	81.64	86.24	60.81	64.59	67.11	69.52	71.61
11	1.039	14.13	25.29	34.85	42.99	60.81	64.59	67.11	69.52	71.61
12	70.05	72.33	76.39	78.15	81.43	60.81	64.59	67.11	69.52	71.61
13	1.224	16.59	29.37	39.47	47.28	60.81	64.59	67.11	69.52	71.61
14	66.33	67.06	72.13	72.60	77.38	60.81	64.59	67.11	69.52	71.61
15	1.341	18.15	31.97	42.54	50.32	60.81	64.59	67.11	69.52	71.61
16	66.80	68.41	71.97	73.36	76.49	60.81	64.59	67.11	69.52	71.61
17	1.415	19.14	33.65	44.61	52.53	60.81	64.59	67.11	69.52	71.61
18	67.12	67.35	71.80	71.90	76.24	60.81	64.59	67.11	69.52	71.61
19	1.460	19.73	34.68	45.96	54.13	60.81	64.59	67.11	69.52	71.61
20	71.27	72.91	75.47	76.77	78.71	60.81	64.59	67.11	69.52	71.61
21	1.481	20.01	35.18	46.63	55.00	60.81	64.59	67.11	69.52	71.61
22	73.87	75.82	77.81	79.27	80.42	60.81	64.59	67.11	69.52	71.61

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

L\*\*3/T

CUMULATIVE VOLUMES

L\*\*3

RATES FOR THIS TIME STEP

IN:	IN:
STORAGE = .0	STORAGE = .0
CONSTANT HEAD = .0	CONSTANT HEAD = .0
WELLS = .0	WELLS = .0
DRAINS = .0	DRAINS = .0
RECHARGE = .13608E+08	RECHARGE = 157.50
TOTAL IN = .13608E+08	TOTAL IN = 157.50
OUT:	OUT:
STORAGE = .0	STORAGE = .0
CONSTANT HEAD = .43265E+07	CONSTANT HEAD = 50.075
WELLS = .64800E+07	WELLS = 75.000
DRAINS = .28011E+07	DRAINS = 32.420
RECHARGE = .0	RECHARGE = .0
TOTAL OUT = .13608E+08	TOTAL OUT = 157.49
IN - OUT = 184.00	IN - OUT = .21210E-02
PERCENT DISCREPANCY = 0.00	PERCENT DISCREPANCY = 0.00

TIME SUMMARY AT END OF TIME STEP 1 IN STRESS PERIOD 1

	SECONDS	MINUTES	HOURS	DAYS	YEARS
TIME STEP LENGTH	86400.0	1440.00	24.0000	1.00000	.273785E-02
STRESS PERIOD TIME	86400.0	1440.00	24.0000	1.00000	.273785E-02
TOTAL SIMULATION TIME	86400.0	1440.00	24.0000	1.00000	.273785E-02

## APPENDIX E

### ABBREVIATED INPUT INSTRUCTIONS

These input instructions are intended as a quick reference for the experienced user. Most explanations that are contained in the complete input instructions given in package documentation have been omitted. The format of input fields is given only for those records that contain fields that are not 10 characters wide. Each input item, for which format is not given, is identified as either a record or an array. For records, the fields contained in the record are named. For arrays, only the array name is given. Input fields which contain codes or flags are described. All other field and array descriptions have been dropped.

#### Array Input

The real two-dimensional array reader (U2DREL), the integer two-dimensional array reader (U2DINT), and the real one-dimensional array reader (U1DREL) read one array-control record and, optionally, a data array in a format specified on the array-control record.

FOR REAL ARRAY READER (U2DREL or U1DREL)

Data:	LOCAT	CNSTNT	FMTIN	IPRN
Format:	I10	F10.0	5A4	I10

FOR INTEGER ARRAY READER (U2DINT)

Data:	LOCAT	ICONST	FMTIN	IPRN
Format:	I10	I10	5A4	I10

IPRN--is a flag indicating that the array being read should be printed and a code for indicating the format that should be used. It is used only if LOCAT is not equal to zero. The format codes are different for each of the three modules. IPRN is set to zero when the specified value exceeds those defined in the chart below. If IPRN is less than zero, the array will not be printed.

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

LOCAT--indicates the location of the data which will be put in the array.

If LOCAT < 0, unit number for unformatted records.

If LOCAT = 0, all elements are set equal to CNSTNT or ICONST.

If LOCAT > 0, unit number for formatted records.

## Basic Package Input

Input for the Basic (BAS) Package except for output control is read from unit 1 as specified in the main program. If necessary, the unit number for BAS input can be changed to meet the requirements of a particular computer. Input for the output control option is read from the unit number specified in IUNIT(12).

### FOR EACH SIMULATION

1. Record: HEADNG(32)
2. Record: HEADNG (continued)
3. Record: NLAY    MROW    NCOL    NPER    ITMUNI
4. Data:    IUNIT(24)  
Format: 24I3  
(BCF WEL DRN RIV EVT XXX GHB RCH SIP XXX SOR OC)  
      1  2  3  4  5  6  7  8  9 10 11 12
5. Record: IAPART    ISTRT
6. Array:    IBOUND(NCOL,NROW)  
(One array for each layer in the grid)
7. Record: HNOFLO
8. Array:    Shead(NCOL,NROW)  
(One array for each layer in the grid)

### FOR EACH STRESS PERIOD

9. Data:    PERLEN    NSTP    TSMULT

ITMUNI--is the time unit of model data.

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

Consistent length and time units must be used for all model data. The user may choose one length unit and one time unit to be used to specify all input data.

IUNIT--is a 24-element table of input units for use by all major options.

IAPART--indicates whether array BUFF is separate from array RHS.

If IAPART = 0, the arrays BUFF and RHS occupy the same space. This option conserves space. This option should be used unless some other package explicitly says otherwise.

If IAPART ≠ 0, the arrays BUFF and RHS occupy different space.

ISTRT--indicates whether starting heads are to be saved.

If ISTRT = 0, starting heads are not saved.

If ISTRT ≠ 0, starting heads are saved.

IBOUND--is the boundary array.

If IBOUND(I,J,K) < 0, cell I,J,K has a constant head.

If IBOUND(I,J,K) = 0, cell I,J,K is inactive.

If IBOUND(I,J,K) > 0, cell I,J,K is active.

HNOFLO--is the value of head to be assigned to all inactive cells.

Shead--is head at the start of the simulation.

PERLEN--is the length of a stress period.

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

TSMULT--is the multiplier for the length of successive time steps.

## Output Control Input

Input to Output Control is read from the unit specified in IUNIT(12). All printer output goes to unit 6 as specified in the main program. If necessary, the unit number for printer output can be changed to meet the requirements of a particular computer.

FOR EACH SIMULATION

1. Record: IHEDFM        IDDNFM        IHEDUN        IDDNUN

FOR EACH TIME STEP

2. Record: INCODE IHDDFL IBUDFL ICBCFL

3. Record: Hdpr    Ddpr    Hdsv    Ddsv

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

IHEDFM--is a code for the format in which heads will be printed.

IDDNFM--is a code for the format in which drawdowns will be printed.

	0 - (10G11.4)	7 - (20F5.0)
	1 - (11G10.3)	8 - (20F5.1)
positive--wrap	2 - (9G13.6)	9 - (20F5.2)
	3 - (15F7.1)	10 - (20F5.3)
negative--strip	4 - (15F7.2)	11 - (20F5.4)
	5 - (15F7.3)	12 - (10G11.4)
	6 - (15F7.4)	

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

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

INCODE--is the head/drawdown output code.

If INCODE < 0, layer-by-layer specifications from the last time steps are used. Input item 3 is not read.

If INCODE = 0, all layers are treated the same way. Input item 3 will consist of one record. IOFLG array will be read.

If INCODE > 0, input item 3 will consist of one record for each layer.

IHDDFL--is a head and drawdown output flag.

If IHDDFL = 0, neither heads nor drawdowns will be printed or saved.

If IHDDFL ≠ 0, heads and drawdowns will be printed or saved.

IBUDFL--is a budget print flag.

If IBUDFL = 0, overall volumetric budget will not be printed.

If IBUDFL ≠ 0, overall volumetric budget will be printed.

ICBCFL--is a cell-by-cell flow-term flag.

If ICBCFL = 0, cell-by-cell flow terms are not saved or printed.

If ICBCFL ≠ 0, cell-by-cell flow terms are printed or recorded on disk depending on flags set in the component of flow packages, i.e., IWELCB, IRCHCB, etc.

Hdpr--is the output flag for head printout.

If Hdpr = 0, head is not printed for the corresponding layer.

If Hdpr ≠ 0, head is printed for the corresponding layer.

Ddpr--is the output flag for drawdown printout.

If Ddpr = 0, drawdown is not printed for the corresponding layer.

If Ddpr ≠ 0, drawdown is printed for the corresponding layer.

Hdsv--is the output flag for head save.

If Hdsv = 0, head is not saved for the corresponding layer.

If Hdsv ≠ 0, head is saved for the corresponding layer.

Ddsv--is the output flag for drawdown save.

If Ddsv = 0, drawdown is not saved for the corresponding layer.

If Ddsv ≠ 0, drawdown is saved for the corresponding layer.

## Block-Centered Flow Package Input

Input for the BCF Package is read from the unit specified in IUNIT(1).

### FOR EACH SIMULATION

1. Record: ISS        IBCFCB
2. Data: LAYCON(NLAY) (maximum of 80 layers)  
Format: 40I2  
(If there are 40 or fewer layers, use one record.)
3. Array: TRPY(NLAY)
4. Array: DELR(NCOL)
5. Array: DELC(NROW)

All of the arrays (items 6-12) for layer 1 are read first; then all of the arrays for layer 2, etc.

### IF THE SIMULATION IS TRANSIENT

6. Array: sf1(NCOL,NROW)

### IF THE LAYER TYPE CODE (LAYCON) IS ZERO OR TWO

7. Array: Tran(NCOL,NROW)

### IF THE LAYER TYPE CODE (LAYCON) IS ONE OR THREE

8. Array: HY(NCOL,NROW)
9. Array: BOT(NCOL,NROW)

### IF THIS IS NOT THE BOTTOM LAYER

10. Array: Vcont(NCOL,NROW)

### IF THE SIMULATION IS TRANSIENT AND THE LAYER TYPE CODE (LAYCON) IS TWO OR THREE

11. Array: sf2(NCOL,NROW)

### IF THE LAYER TYPE CODE IS TWO OR THREE

12. Array: TOP(NCOL,NROW)

ISS--is the steady-state flag.

If ISS  $\neq$  0, the simulation is steady state.

If ISS = 0, the simulation is transient.

IBCFCB--is a flag and a unit number.

If IBCFCB > 0, cell-by-cell flow terms will be recorded if ICBCFL (see Output Control) is set.

If IBCFCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IBCFCB < 0, print flow for constant-head cells if ICBCFL is set.

LAYCON--is the layer type table: 0 - confined, 1 - unconfined,

2 - confined/unconfined (T constant), and 3 - confined/unconfined.

TRPY--is an anisotropy factor for each layer: T or K along a column to T or K along a row.

DELR--is the cell width along rows.

DELC--is the cell width along columns.

sf1--is the primary storage factor.

Tran--is the transmissivity along rows.

HY--is the hydraulic conductivity along rows.

BOT--is the elevation of the aquifer bottom.

Vcont--is the vertical hydraulic conductivity divided by the thickness from a layer to the layer beneath it.

sf2--is the secondary storage factor.

TOP--is the elevation of the aquifer top.

### River Package Input

Input to the River (RIV) Package is read from the unit specified in IUNIT(4).

FOR EACH SIMULATION

1. Record: MXRIVR IRIVCB

FOR EACH STRESS PERIOD

2. Record: ITMP

3. Record: Layer Row Column Stage Cond Rbot  
(Input item 3 normally consists of one record for each river reach. If ITMP is negative or zero, item 3 is not read.)

IRIVCB--is a flag and a unit number.

If IRIVCB > 0, cell-by-cell flow terms will be recorded.

If IRIVCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IRIVCB < 0, river leakage will be printed if ICBCFL is set.

ITMP--is a flag and a counter.

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

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

### Recharge Package Input

Input to the Recharge (RCH) Package is read from the unit specified in IUNIT(8).

FOR EACH SIMULATION

1. Record: NRCHOP IRCHCB

FOR EACH STRESS PERIOD

2. Record: INRECH INIRCH

3. Array: RECH(NCOL,NROW)

IF THE RECHARGE OPTION IS EQUAL TO 2

4. Array: IRCH(NCOL,NROW)

NRCHOP--is the recharge option code.

1 - Recharge is only to the top grid layer.

2 - Vertical distribution of recharge is specified in array IRCH.

3 - Recharge is applied to the highest active cell in each vertical column.

IRCHCB--is a flag and a unit number.

If IRCHCB > 0, unit number for cell-by-cell flow terms.

If IRCHCB  $\leq$  0, cell-by-cell flow terms will not be printed or recorded.

INRECH--is the RECH read flag.

If INRECH < 0, recharge rates from the preceding stress period are used.

If INRECH  $\geq$  0, an array of recharge rates, (RECH) is read.

INIRCH--is similar to INRECH.

### Well Package Input

Input for the Well (WEL) Package is read from the unit specified in IUNIT(2).

FOR EACH SIMULATION

1. Record: MXWELL IWELCB

FOR EACH STRESS PERIOD

2. Record: ITMP

3. Record: Layer Row Column Q

(Input item 3 normally consists of one record for each well. If ITMP is negative or zero, item 3 is not read.)

MXWELL--is the maximum number of wells used at any time.

IWELCB--is a flag and a unit number.

If IWELCB > 0, unit number for cell-by-cell flow terms.

If IWELCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IWELCB < 0, well recharge will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

### Drain Package Input

Input to the Drain (DRN) Package is read from the unit specified in IUNIT(3).

FOR EACH SIMULATION

1. Record: MXDRN IDRNCB

FOR EACH STRESS PERIOD

2. Record: ITMP

3. Record: Layer Row Col Elevation Cond

(Input item 3 normally consists of one record for each drain.

If ITMP is negative or zero, item 3 will not be read.)

MXDRN--is the maximum number of drain cells active at one time.

IDRNCB--is a flag and a unit number.

If IDRNCB > 0, unit number for cell-by-cell flow terms.

If IDRNCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IDRNCB < 0, drain leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

## Evapotranspiration Package Input

Input to the Evapotranspiration (EVT) Package is read from the unit specified in IUNIT (5).

FOR EACH SIMULATION

1. Record: NEVTOP      IEVTCB

FOR EACH STRESS PERIOD

2. Record: INSURF      INEVTR      INEXDP      INIEVT

3. Array: SURF

4. Array: EVTR

5. Array: EXDP

IF THE ET OPTION IS EQUAL TO TWO

6. Array: IEVT

NEVTOP--is the evapotranspiration (ET) option code.

1 - ET is calculated only for cells in the top grid layer.

2 - The cell for each vertical column is specified by the user in array IEVT.

IEVTCB--is a flag and a unit number.

If IEVTCB > 0, unit number for cell-by-cell flow terms.

If IEVTCB ≤ 0, cell-by-cell flow terms will not be printed or recorded.

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

If INSURF ≥ 0, an array containing the ET surface elevation will be read.

If INSURF < 0, the ET surface from the preceding stress period will be reused.

INEVTR--is similar to INSURF.

INEXDP--is similar to INSURF.

INIEVT--is similar to INSURF.

### General-Head Boundary Package Input

Input for the General-Head Boundary (GHB) Package is read from the unit specified in IUNIT(7).

FOR EACH SIMULATION

1. Record: MXBND IGHBCB

FOR EACH STRESS PERIOD

2. Record: ITMP Boundary

3. Record: Layer Row Column Head Cond

(Input item 3 normally consists of one record for each GHB.

If ITMP is negative or zero, item 3 is not read.)

MXBND--is the maximum number of general-head boundary cells at one time.

IGHBCB--is a flag and a unit number.

If IGHBCB > 0, unit number for cell-by-cell flow terms.

If IGHBCB = 0, cell-by-cell flow terms will not be printed or recorded.

If IGHBCB < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

### Strongly Implicit Procedure Package Input

Input to the Strongly Implicit Procedure (SIP) Package is read from the unit specified in IUNIT(9).

FOR EACH SIMULATION

1. Record: MXITER NPARM

2. Record: ACCL HCLOSE IPCALC WSEED IPRSIP

IPCALC--is a flag indicating where the iteration parameter seed will come from.

0 - the seed will be entered by the user.

1 - the seed will be calculated at the start of the simulation from problem parameters.

IPRSIP--is the printout interval for SIP.

### Slice-Successive Overrelaxation Package Input

Input to the Slice-Successive Overrelaxation (SOR) Package is read from the unit specified in IUNIT(11).

FOR EACH SIMULATION

1. Record: MXITER

2. Record: ACCL HCLOSE IPRSOR

IPRSOR--is the printout interval for SOR.

