Downey
H

NNA.870729.0113

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

.'ED 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

Page

ILLUSTRATIONS

TABLES

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 programing structure wherein similar programing 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.[1]/

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

[1]"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 Implemementation 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

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

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 system consisting of "nrow" rows, "ncol" columns, and "nlay" layers, i is the row index, i = 1,2,. . . nrow; j is the column index, j = 1,2,. . . ncol; and k is the layer index, k = 1,2,. . . 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 Δr_j for the jth column; the width of cells along columns are designated as Δc_i for the ith row; and the thickness of layers in the vertical are designated as Δv_k for the kth 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 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

$$\Sigma Q_i = S_S \frac{\Delta h}{\Delta t} \Delta V \qquad (2)$$

where

Q_i 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});

ΔV is the volume of the cell (L^3); and

Δh is the change in head over a time interval of length Δt.

The term on the right hand side is equivalent to the volume of water taken into storage over a time interval Δt given a change in head of Δh.

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

$$q_{i,j-1/2,k} = KR_{i,j-1/2,k} \Delta c_i \Delta v_k \frac{(h_{i,j-1,k} - h_{i,j,k})}{\Delta r_{j-1/2}} \qquad (3)$$

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,

$$q_{i,j+1/2,k} = KR_{i,j+1/2,k} \Delta c_i \Delta v_k \frac{(h_{i,j+1,k} - h_{i,j,k})}{\Delta r_{j+1/2}} \qquad (4)$$

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

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

and flow through the rear face of the block is

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

For the vertical direction, flow through the bottom face is

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

while flow through the upper face is given by

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

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,

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

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

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

Similarly, equations 4-8 can be rewritten to yield

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

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

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

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

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

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

$$a_{i,j,k,r} = p_{i,j,k,n}h_{i,j,k} + q_{i,j,k,n} \tag{16}$$

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,

$$a_{i,j,k,1} = q_{i,j,k,1}. \tag{17}$$

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

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

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

$$a_{i,j,k,2} = - CRIV_{i,j,k,2}h_{i,j,k} + CRIV_{i,j,k,2}R_{i,j,k}. \tag{19}$$

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

$$QS_{i,j,k} = \sum_{n=1}^{N} a_{i,j,k,n} = \sum_{n=1}^{N} p_{i,j,k,n} \; h_{i,j,k} + \sum_{n=1}^{N} q_{i,j,k,n}. \tag{20}$$

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

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

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

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

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

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

$$q_{i,j-1/2,k} + q_{i,j+1/2,k} + q_{i-1/2,j,k} + q_{i+1/2,j,k}$$
$$+ q_{i,j,k-1/2} + q_{i,j,k+1/2} + QS_{i,j,k} = SS_{i,j,k}\frac{\Delta h_{i,j,k}}{\Delta t}\Delta r_j \Delta c_i \Delta v_k \tag{22}$$

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

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

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, $h^m_{i,j,k}$ and $h^{m-1}_{i,j,k}$, are indicated on the vertical axis; the slope of the dotted line is $\Delta h^m_{i,j,k}/\Delta t_m$. 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

$$\frac{\Delta h^m_{i,j,k}}{\Delta t_m} = \frac{h^m_{i,j,k} - h^{m-1}_{i,j,k}}{t_m - t_{m-1}}. \tag{24}$$

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--

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,

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

In this equation, head at the beginning of the time step $h^m_{i,j,k}$ 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

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

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 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

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

where

$$HCOF_{i,j,k} = P_{i,j,k} - SC1_{i,j,k}/(t_m - t_{m-1}); \qquad (L^2t^{-1})$$

$$RHS_{i,j,k} = - Q_{i,j,k} - SC1_{i,j,k}h^{m-1}_{i,j,k}/(t_m - t_{m-1}); \text{ and} \qquad (L^3t^{-1})$$

$$SC1_{i,j,k} = SS_{i,j,k}\Delta r_j \Delta c_i \Delta v_k. \qquad (L^2)$$

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 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.

Package Name	Abbreviation	Package Description
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.
Evapotrans-piration	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 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

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 contant-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.

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 ft^2/day and pumpage as m^3/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.

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.

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 (Δv_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.

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

$$
\begin{aligned}
&CR_{i,j-1/2,k}(h^m_{i,j-1,k}-h^m_{i,j,k}) + CR_{i,j+1/2,k}(h^m_{i,j+1,k} - h^m_{i,j,k}) \\
&+ CC_{i-1/2,j,k}(h^m_{i-1,j,k}-h^m_{i,j,k}) + CC_{i+1/2,j,k}(h^m_{i+1,j,k} - h^m_{i,j,k}) \\
&+ CV_{i,j,k-1/2}(h^m_{i,j,k-1}-h^m_{i,j,k}) + CV_{i,j,k+1/2}(h^m_{i,j,k+1}-h^m_{i,j,k}) \\
&+ P_{i,j,k}h^m_{i,j,k}+Q_{i,j,k} = SC1_{i,j,k}\,(h^m_{i,j,k}-h^{m-1}_{i,j,k})/\Delta t_m. \qquad (28)
\end{aligned}
$$

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 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.

```
       VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP  1 IN STRESS PERIOD  1
       ----------------------------------------------------------------------------
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 =    15
       TOTAL IN =     .13608E+08                                        TOTAL IN =    15
     OUT:                                                             OUT:
     ----                                                             ----
        STORAGE =     .0                                                 STORAGE =     .0
  CONSTANT HEAD =     .43265E+07                                   CONSTANT HEAD =    50.
          WELLS =     .64800E+07                                           WELLS =    75.
         DRAINS =     .28010E+07                                          DRAINS =    32.
       RECHARGE =     .0                                                RECHARGE =     .0
      TOTAL OUT =     .13607E+08                                       TOTAL OUT =    15
       IN - OUT =     303.00                                            IN - OUT =     .3
PERCENT DISCREPANCY =                0.00                        PERCENT DISCREPANCY =
```

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.

IUNIT LOCATION	MAJOR OPTION
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 ≠ 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.

ISTRT--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 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 (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

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

SAMPLE INPUT TO THE BASIC PACKAGE(EXCLUDING OUTPUT CONTROL)

```
DATA        EXPLANATION                                          INPUT RECORDS
ITEM

 1    (HEADNG)___________________________________________   SAMPLE INPUT FOR BASIC PACKAGE 3 LAYERS 15 ROWS 15
 2    (HEADNG) CONTINUED_________________________________   WITH RECHARGE, DRAINS, AND WELLS
 3    (NLAY, NROW, NCOL, NPER, ITMUNI)___________________             3          10          12          2        1
 4    (IUNIT TABLE)______________________________________    31 12 14 00 00 00 00 28 19 00 00 22
 5    (IAPART, ISTRT)____________________________________             0          1
 6    CONTROL RECORD FOR IBOUND ARRAY LAYER 1____________             1          1 (12I2)                          3
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
         IBOUND VALUES FOR LAYER 1                          -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
                                                            -1 1 1 1 1 1 1 1 1 1 1 1 1 1
 6    CONTROL RECORD FOR IBOUND ARRAY LAYER 2____________             1          1 (12I3)                          3
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
         IBOUND VALUES FOR LAYER 2                           -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
                                                             -2  2  2  2  2  2  2  2  2  2  2  2
 6    CONTROL RECORD FOR IBOUND ARRAY LAYER 3____________              0          3
 7    (HNOFLO)___________________________________________        999.99
 8    CONTROL RECORD FOR STARTING HEAD ARRAY FOR LAYER 1_             1          1 (12F5.1)                        6
                                                              10.  17.  24.  31.  38.  45.  52.  59.  66.  73.
                                                              11.  19.  27.  35.  43.  51.  59.  67.  75.  83.
                                                              13.  23.  33.  43.  53.  63.  73.  83.  93. 103.
                                                              16.  29.  42.  55.  68.  81.  94. 107. 120. 133.
      STARTING HEAD VALUES FOR LAYER 1                        20.  37.  54.  71.  88. 105. 122. 139. 156. 173.
                                                              20.  37.  54.  71.  88. 105. 122. 139. 156. 173.
                                                              16.  29.  42.  55.  68.  81.  94. 107. 120. 133.
                                                              13.  23.  33.  43.  53.  63.  73.  83.  93. 103.
                                                              11.  19.  27.  35.  43.  51.  59.  67.  75.  83.
                                                              10.  17.  24.  31.  38.  45.  52.  59.  66.  73.
 8    CONTROL RECORD FOR STARTING HEAD ARRAY FOR LAYER 2_            23          1 (12F6.2)                        5
 8    CONTROL RECORD FOR STARTING HEAD ARRAY FOR LAYER 3_             0        0.0
 9    STRESS PERIOD 1----(PERLEN  NSTP  TSMULT)__________         86400          5        1.5
 9    STRESS PERIOD 2----(PERLEN  NSTP  TSMULT)__________        172800          7        1.5
```

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 (ISTRT 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

BAS1OC

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 ouput 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 ≠ 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.

Module Documentation for the Basic Package

The Basic Package (BAS1) consists of eight primary modules and five submodules. The modules are:

Primary Modules

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

Submodules

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

Narrative for Module BAS1DF

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

1. Print the name of the program.

2. Read and print a heading.

3. Read the number of layers, rows, columns, stress periods, and units of time code ITMUNI. ITMUNI is a code which indicates the time units of model data. It does not affect model calculations but is used when printing the amount of elapsed time (see the input instructions for the codes).

4. Print the number of layers, rows, columns, and stress periods.

5. Select and print a message showing the time units.

6. Read and print the input unit numbers IUNIT for all major options. IUNIT is a 24-element table. Each entry has been assigned to a particular major option. The user specifies that a certain major option is to be used by putting a positive integer into the IUNIT entry corresponding to that major option. The integer is the unit number from which input to the major option will be read. If a major option is not going to be used, the corresponding IUNIT element is set equal to zero.

7. Initialize the total-elapsed time counter (TOTIM) and the storage-array counter (ISUM) and calculate the total number of cells.

8. RETURN.

List of Variables for Module BAS1DF

Variable	Range	Definition
I	Module	Index.
INBAS	Package	Primary unit number from which input to the BAS1 Package will be read. INBAS = 1.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ITMUNI	Package	Code for time units for this problem: 0 - undefined 1 - seconds 2 - minutes 3 - hours 4 - days 5 - years
IUNIT	Module	DIMENSION (24), Primary input units for each of the major options.
HEADNG	Package	DIMENSION (32), Heading printed on output to identify the problem.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NODES	Global	Number of cells (nodes) in the finite-difference grid.
NPER	Global	Number of stress periods.
NROW	Global	Number of rows in the grid.
TOTIM	Package	Elapsed time in the simulation.

Narrative for Module BAS1AL

Module BAS1AL allocates space for data arrays used by the BAS Package. Space is allocated for HNEW, HOLD, IBOUND, CR, CC, CV, HCOF, RHS, DELR, DELC, and IOFLG. Space is allocated for the STRT array if the user intends to calculate drawdown. Space is also allocated for an array called BUFFER, which is used to accumulate various data arrays such as drawdown and cell-by-cell flow terms when they are being calculated prior to output. To conserve space, the user may specify that arrays BUFFER and RHS should occupy the same space.

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

Module BAS1AL performs its functions in the following order:

1. Print a message identifying the package.

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

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

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

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

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

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

8. RETURN.

List of Variables for Module BAS1AL

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

Narrative for Module BAS1RP

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

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

The user must specify a head value HNOFLO that he wants printed for no-flow (inactive) cells. That value is only used during printing and makes inactive cells stand out on the listing (e.g., 0.C 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.

List of Variables for Module BAS1RP

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

Narrative for Module BAS1ST

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

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

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

Module BAS1ST performs its functions in the following order:

1. Read the length of the stress period (PERLEN), the number of time steps in the stress period (NSTP), and the time-step multiplier (TSMULT).

2. Calculate the length of the first time step.

   (a) Assume the time-step multiplier is equal to one.

   (b) If the time-step multiplier (TSMULT) is not equal to one, calculate the first term of the geometric progression.

3. Print the timing information.

4. Initialize the variable PERTIM which keeps track of elapsed time within a stress period.

5. RETURN.

```
      SUBROUTINE BASIST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C
C-----VERSION 1614 08SEP1982 BASIST
C     ******************************************************************
C     SETUP TIME PARAMETERS FOR NEW TIME PERIOD
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------READ LENGTH OF STRESS PERIOD, NUMBER OF TIME STEPS AND.
C1------TIME STEP MULTIPLIER.
      READ (INBAS,1) PERLEN,NSTP,TSMULT
    1 FORMAT(F10.0,I10,F10.0)
C
C2------CALCULATE THE LENGTH OF THE FIRST TIME STEP.
C
C2A-----ASSUME TIME STEP MULTIPLIER IS EQUAL TO ONE.
      DELT=PERLEN/FLOAT(NSTP)
C
C2B-----IF TIME STEP MULTIPLIER IS NOT ONE THEN CALCULATE FIRST
C2B-----TERM OF GEOMETRIC PROGRESSION.
      IF(TSMULT.NE.1.) DELT=PERLEN*(1.-TSMULT)/(1.-TSMULT**NSTP)
C
C3------PRINT TIMING INFORMATION.
      WRITE (IOUT,2) KPER,PERLEN,NSTP,TSMULT,DELT
    2 FORMAT(1H1,51X,'STRESS PERIOD NO.',I4,', LENGTH =',G15.7/52X
     1,46('-'),/52X,'NUMBER OF TIME STEPS =',I6
     2//53X,'MULTIPLIER FOR DELT =',F10.3
     3//50X,'INITIAL TIME STEP SIZE =',G15.7)
C
C4------INITIALIZE PERTIM (ELAPSED TIME WITHIN STRESS PERIOD).
      PERTIM=0.
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module BAS1ST

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

Narrative for Module BAS1AD

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

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

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

Module BAS1AD performs its functions in the following order:

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

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

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

4. RETURN.

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

List of Variables for Module BAS1AD

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

Narrative for Module BAS1FM

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

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

Module BAS1FM performs its functions in the following order:

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

2. RETURN.

```
      SUBROUTINE BAS1FM(HCOF,RHS,NCOL,NROW,NLAY,NODES)
C
C
C     ******************************************************************
C     SET HCOF=RHS=0.
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION HCOF(NODES),RHS(NODES)
C     ------------------------------------------------------------------
C
C1------FOR EACH CELL INITIALIZE HCOF AND RHS ACCUMULATORS.
      DO 100 I=1,NODES
      HCOF(I)=0.
      RHS(I)=0.
  100 CONTINUE
C
C2------RETURN
      RETURN
      END
```

List of variables for Module BAS1FM

Variable	Range	Definition
I	Module	Index.
HCOF	Global	DIMENSION (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 BAS1OC

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

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

Module BAS1OC performs its functions in the following order:

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

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

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

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

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

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

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

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

9. RETURN.

List of Variables for Module BAS1OC

Variable	Range	Definition
I	Module	Index.
IBUDFL	Package	Flag. = 0, volumetric budget will not be printed for the current time step. ≠ 0, volumetric budget should be printed for the current time step.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be recorded or printed for the current time step.
ICNVG	Global	Flag is set equal to 1 when the iteration procedure has converged.
IHDDFL	Package	Flag. = 0, neither head nor drawdown will be printed at this time step. ≠ 0, head and drawdown may be printed at the end of the current time step.
INCODE	Module	Code specified by user. < 0, reuse contents of IOFLG from the last time step. = 0, read IOFLG for layer 1 and set all other layers to the same thing. > 0, read IOFLG contents for each layer.
INOC	Package	Unit number from which input to output control option will be read.
IOFLG	Package	DIMENSION (NLAY,4), Flags to control printing and recording of head and drawdown for each layer. (NLAY,1) ≠ 0, heads will be printed. (NLAY,2) ≠ 0, drawdown will be printed. (NLAY,3) ≠ 0, heads will be recorded. (NLAY,1) ≠ 0, drawdown will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISTRT	Package	Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
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 BAS1OT

Module BAS1OT 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 BAS1OC (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 (BAS1OT) 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 BAS1OT 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.

```
      SUBROUTINE BAS1OT(HNEW,STRT,ISTRT,BUFF,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 BAS1OT
C     ******************************************************************
C     OUTPUT TIME, VOLUMETRIC BUDGET, HEAD, AND DRAWDOWN
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
     1          VBNM(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,ISTRT,IBOUND,IPFLG,PERTIM,TOTIM)
C
C4------PRINT TOTAL BUDGET IF REQUESTED
  100 IF(IBUDFL.EQ.0) GO TO 120
      CALL SBAS1V(MSUM,VBNM,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 BAS1OT

Variable	Range	Definition
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,1) ≠ 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 BAS1OT (Continued)

Variable	Range	Definition
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 BAS1OT 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.

List of Variables for Module SBAS1D

Variable	Range	Definition
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,1) ≠ 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 BAS1OT 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.

List of Variables for Module SBAS1H

Variable	Range	Definition
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,1) ≠ 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.

List of Variables for Module SBAS1I

Variable	Range	Definition
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.
ISTRT	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.

List of Variables for Module SBAS1T

Variable	Range	Definition
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.

```
      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     ------------------------------------------------------------------
      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

Variable	Range	Definition
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

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

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

$$Q = KA(h_2-h_1)/L \tag{30}$$

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

$$C = KA/L. \tag{31}$$

Therefore, Darcy's law can be written as

$$Q = C(h_2-h_1). \tag{32}$$

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

$$C = TW/L \tag{33}$$

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).

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).

$$C = Q/(h_A - h_B). \tag{34}$$

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

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

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

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

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

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

By comparison with equation 34, it can be seen that

$$\frac{1}{C} = \sum_{i=1}^{n} \frac{1}{C_i}. \tag{38}$$

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

$$C = C_1C_2/(C_1 + C_2). \tag{39}$$

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

$$CR_{i,j+1/2,k} = C_1C_2/(C_1 + C_2). \tag{40}$$

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

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

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

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

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

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

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

if $HNEW_{i,j,k} \geq TOP_{i,j,k}$,

$$\text{then } TR_{i,j,k} = (TOP_{i,j,k} - BOT_{i,j,k})\, HYR_{i,j,k}. \qquad (43)$$

if $TOP_{i,j,k} > HNEW_{i,j,k} > BOT_{i,j,k}$,

$$\text{then } TR_{i,j,k} = (HNEW_{i,j,k} - BOT_{i,j,k})\, HYR_{i,j,k} \qquad (44)$$

if $HNEW_{i,j,k} \leq BOT_{i,j,k}$,

$$\text{then } TR_{i,j,k} = 0 \qquad (45)$$

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

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

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

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

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 and its use unnecessarily complex. The complexity is avoided by allowing the user to calculate Vcont 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 Vcont is to directly use the definition of Vcont (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.

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

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

$$Vcont_{i,j,k+1/2} = \frac{KV_{i,j,k+1/2}}{DELV_{i,j,k+1/2}} = \frac{2}{\dfrac{DELV_{i,j,k}}{KV_{i,j,k}} + \dfrac{DELV_{i,j,k+1}}{KV_{i,j,k+1}}} \qquad (49)$$

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

$$Vcont = 2KV_{i,j,k}/DELV_{i,j,k}. \qquad (50)$$

That is, the low Vcont of a confining bed may dominate the calculation so that Vcont of the aquifer can be ignored.

A third way to calculate Vcont 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 of the equivalent conductance between the two nodes (two aquifer blocks in series with the confining bed) gives

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

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

$$1/CV_{i,j,k+1/2} = 1/C_2 \tag{52}$$

or

$$CV_{i,j,k+1/2} = C_2. \tag{53}$$

Thus the vertical hydraulic conductivity Vcont 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

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

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

$$q_{i,j,k+1/2} = CV_{i,j,k+1/2}(h^m_{i,j,k+1} - h^m_{i,j,k}). \tag{55}$$

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

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

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

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

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, $h^{m}_{i,j,k+1}$ in equation 57 with the head from the previous iteration $h^{m-1}_{i,j,k+1}$ gives an approximation to q.

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

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 $CV_{i,j,k+1/2}(TOP_{i,j,k+1} - h^{m-1}_{i,j,k+1})$ 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

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

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

$$Q = SA(HOLD_{i,j,k} - HNEW_{i,j,k})/DELT \qquad (60)$$

where

A is the area of the cell $DELR_j$ times $DELC_i$ (L^2);

DELT 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

$$Q = (ST1(HOLD-TOP)+ST2(TOP-HNEW))A/(DELT) \tag{61}$$

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

$$Q = (-SNEW \times HNEW) + (SOLD(HOLD - TOP) + SNEW \times TOP) \tag{62}$$

where

SNEW is ST2 x A/DELT (L^2t^{-1}); and

SOLD is ST1 x A/DELT (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 programing 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 ≠ 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.

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.

```
      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     ------------------------------------------------------------------
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
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

Variable	Range	Definition
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 be 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 U1DREL 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 U1DREL 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.

```
      SUBROUTINE BCF1RP(IBOUND,HNEW,SC1,HY,CR,CC,CV,DELR,DELC,
     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     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NODES),SC1(NODES),HY(NODES),CR(NODES),CC(NODES),
     1          CV(NODES),ANAME(6,10),DELR(NCOL),DELC(NROW),BOT(NODES),
     1          TOP(NODES),SC2(NODES),TRPY(NLAY),IBOUND(NODES)
C
      COMMON /FLWCOM/LAYCON(80)
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,4HS. 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    ,4HDELR/
      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    ,4HDELC/
C     ------------------------------------------------------------------
C
C1------CALCULATE NUMBER OF NODES IN A LAYER AND READ TRPY,DELR,DELC
      NIJ=NCOL*NROW
C
      CALL U1DREL(TRPY,ANAME(1,8),NLAY,IN,IOUT)
      CALL U1DREL(DELR,ANAME(1,9),NCOL,IN,IOUT)
      CALL U1DREL(DELC,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

Variable	Range	Definition
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)

Variable	Range	Definition
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.

<u>List of Variables for Module BCF1FM</u>

Variable	Range	Definition
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)

Variable	Range	Definition
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 BAS1OT 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.

```
      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     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
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
      COMMON /FLWCOM/LAYCON(80)
C
      DIMENSION TEXT(4)
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ',' STO','RAGE'/
C     ------------------------------------------------------------------
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

Variable	Range	Definition
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)

Variable	Range	Definition
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.

Code	Status
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 capcacity (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.

List of Variables for Module SBCF1N

Variable	Range	Definition
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.
HCNV	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.

List of Variables for Module SBCF1H

Variable	Range	Definition
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

$$C = C_1C_2/(C_1 + C_2).$$

However, C_1 and C_2 can be represented by

$$C_1 = T_1W/(L_1/2)$$

$$C_2 = T_2W/(L_2/2).$$

Thus,

$$C = 2T_1T_2W/(T_1L_2 + T_2L_1).$$

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.

```
      SUBROUTINE SBCF1C(CR,CC,TRPY,DELR,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
      DIMENSION CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY)
     2   , TRPY(NLAY), DELR(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.
      DO 40 I=1,NROW
      DO 40 J=1,NCOL
      T1=CC(J,I,K)
C
C2------IF T=0 THEN SET CONDUCTANCE EQUAL TO 0. GO ON TO NEXT CELL.
      IF(T1.NE.0.) GO TO 10
      CR(J,I,K)=0.
      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.
   10 IF(J.EQ.NCOL) GO TO 30
      T2=CC(J+1,I,K)
      CR(J,I,K)=2.*T2*T1*DELC(I)/(T1*DELR(J+1)+T2*DELR(J))
C
C4------IF THIS IS NOT THE LAST ROW(FRONTMOST) THEN CALCULATE
C4------BRANCH CONDUCTANCE IN THE COLUMN DIRECTION (CC) TO THE FRONT.
   30 IF(I.EQ.NROW) GO TO 40
      T2=CC(J,I+1,K)
      CC(J,I,K)=YX*T2*T1*DELR(J)/(T1*DELC(I+1)+T2*DELC(I))
   40 CONTINUE
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module SBCF1C

Variable	Range	Definition
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.

```
      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     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,HD
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
      COMMON /FLWCOM/LAYCON(80)
C
      DIMENSION TEXT(12)
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
      NCM1=NCOL-1
      IF(NCM1.LT.1) GO TO 405
C
C1-----CLEAR THE BUFFER
      DO 310 K=1,NLAY
      DO 310 I=1,NROW
      DO 310 J=1,NCOL
      BUFF(J,I,K)=0.
  310 CONTINUE
C
C2-----FOR EACH CELL CALCULATE FLOW THRU RIGHT FACE & STORE IN BUFFER
      DO 400 K=1,NLAY
      DO 400 I=1,NROW
      DO 400 J=1,NCM1
      IF((IBOUND(J,I,K).LE.0) .AND. (IBOUND(J+1,I,K).LE.0)) GO TO 400
      HDIFF=HNEW(J,I,K)-HNEW(J+1,I,K)
      BUFF(J,I,K)=HDIFF*CR(J,I,K)
  400 CONTINUE
C
C3-----RECORD CONTENTS OF BUFFER
      CALL UBUDSV(KSTP,KPER,TEXT(1),IBCFCB,BUFF,NCOL,NROW,NLAY,IOUT)
C
C4-----CLEAR THE BUFFER
  405 NRM1=NROW-1
      IF(NRM1.LT.1) GO TO 505
      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

Variable	Range	Definition
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.
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 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.

List of Variables for Module SBCF1F

Variable	Range	Definition
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.
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.
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)

Variable	Range	Definition
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).
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

$$QRIV = KLW(HRIV - HAQ)/M \tag{63}$$

where

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

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

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

$$QRIV = CRIV(HRIV - HAQ) \tag{64}$$

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

$$QRIV = CRIV(HRIV - H) \tag{65}$$

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 (RBOT). In that case, equation 64 can be written

$$QRIV = CRIV(HRIV - RBOT). \tag{66}$$

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.

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.

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 RIV1AL

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.

```
      SUBROUTINE RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,IN,IOUT,
     1                        IRIVCB)
C
C-----VERSION 0935 08DEC1983 RIV1AL
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

Variable	Range	Definition
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.

List of Variables for Module RIV1RP

Variable	Range	Definition
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.

```
      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
      DOUBLE PRECISION HNEW
      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

Variable	Range	Definition
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.

15. 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 BAS1OT. Add RATIN and RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS1OT. Move the river budget term labels to VBNM for printing by BAS1OT.

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

19. RETURN.

```
      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     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
      DIMENSION RIVR(6,MXRIVR),IBOUND(NCOL,NROW,NLAY),
     1          HNEW(NCOL,NROW,NLAY),VBVL(4,20),VBNM(4,20),
     2          BUFF(NCOL,NROW,NLAY)
      DIMENSION TEXT(4)
      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

Variable	Range	Definition
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)

Variable	Range	Definition
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

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

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.

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 ≤ 0, cell-by-cell flow terms will not be printed or recorded.

INRECH--is the RECH read flag.

If INRECH ≥ 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 ≥ 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.

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.

List of Variables for Module RCH1AL

Variable	Range	Definition
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.
IRCHCB	Package	Flag. IRCHCB ≤ 0, cell-by-cell flow terms will not be printed or recorded. IRCHCB > 0 and ICBCFL ≠ 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	Module	ISUM-1.
LCIRCH	Package	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	Global	Number of columns 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.

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 perioo 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.

```
      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     ------------------------------------------------------------------
      DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
     1          ANAME(6,2),DELR(NCOL),DELC(NROW)
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'/
      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

Variable	Range	Definition
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. $\geq$ 0, IRCH array will be read. < 0, IRCH array already in memory from the last stress period will be used.
INRECH	Module	Flag. $\geq$ 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

List of Variables for Module RCH1FM

Variable	Range	Definition
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.