USGS
science for a changing world

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

Chapter A1

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

By Michael G. McDonald and
Arlen W. Harbaugh

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

Book 6

MODELING TECHNIQUES

DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, *Secretary*

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, *Director*

A copy of the source program of the Modular Model
is available at cost, from:
Chief, Office of Ground Water, MS 411, National Center,
U.S. Geological Survey, Reston, Virginia 22092

UNITED STATES GOVERNMENT PRINTING OFFICE, WASHINGTON : 1988

For sale by the Books and Open-File Reports Section, U.S. Geological Survey,
Federal Center, Box 25425, Denver, CO 80225

PREFACE

The series of manuals on techniques describes procedures for planning and executing specialized work in water-resources investigations. The material is grouped under major subject headings called books and further subdivided into sections and chapters. Section A of Book 6 is on ground-water modeling.

Provisional drafts of chapters are distributed to field offices of the U.S. Geological Survey for their use. These drafts are subject to revision because of experience in use or because of advancement in knowledge, techniques, or equipment. After the technique described in a chapter is sufficiently developed, the chapter is published and is for sale by the Books and Open-File Reports Section, U.S. Geological Survey, Federal Center, Box 25425, Denver, Colorado 80225.

Reference to trade names, commercial products, manufacturers, or distributors in this manual constitutes neither endorsement by the U.S. Geological Survey nor recommendation for use.

TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS OF THE UNITED STATES GEOLOGICAL SURVEY

The U.S. Geological Survey publishes a series of manuals describing procedures for planning and conducting specialized work in water-resources investigations. The manuals published to date are listed below and may be ordered by mail from the **U.S. Geological Survey, Books and Open-File Reports, Federal Center, Box 25425, Denver, Colorado 80225** an authorized agent of the Superintendent of Documents, Government Printing Office).

Prepayment is required. Remittance should be sent by check or money order payable to U.S. Geological Survey. Prices are not included in the listing below as they are subject to change. **Current prices can be obtained** by writing to the USGS, Books and Open File Reports. Prices include cost of domestic surface transportation. For transmittal outside the U.S.A. (except to Canada and Mexico) a surcharge of 25 percent of the net bill should be included to cover surface transportation. When ordering any of these publications, please give the title, book number, chapter number, and "U.S. Geological Survey Techniques of Water-Resources Investigations."

TWI 1-D1. Water temperature—influential factors, field measurement, and data presentation, by H.H. Stevens, Jr., J.F. Ficke, and G.F. Smoot, 1975, 65 pages.
TWI 1-D2. Guidelines for collection and field analysis of ground-water samples for selected unstable constituents, by W.W. Wood. 1976. 24 pages.
TWI 2-D1. Application of surface geophysics to ground water investigations, by A.A.R. Zohdy, G.P. Eaton, and D.R. Mabey. 1974. 116 pages.
TWI 2-E1. Application of borehole geophysics to water-resources investigations, by W.S. Keys and L.M. MacCary. 1971. 126 pages.
TWI 3-A1. General field and office procedures for indirect discharge measurement, by M.A. Benson and Tate Dalrymple. 1967. 30 pages.
TWI 3-A2. Measurement of peak discharge by the slope-area method, by Tate Dalrymple and M.A. Benson. 1967. 12 pages.
TWI 3-A3. Measurement of peak discharge at culverts by indirect methods, by G.L. Bodhaine. 1968. 60 pages.
TWI 3-A4. Measurement of peak discharge at width contractions by indirect methods, by H.F. Matthai. 1967. 44 pages.
TWI 3-A5. Measurement of peak discharge at dams by indirect methods, by Harry Hulsing. 1967. 29 pages.
TWI 3-A6. General procedure for gaging streams, by R.W. Carter and Jacob Davidian. 1968. 13 pages.
TWI 3-A7. Stage measurements at gaging stations, by T.J. Buchanan and W.P. Somers. 1968. 28 pages.
TWI 3-A8. Discharge measurements at gaging stations, by T.J. Buchanan and W.P. Somers. 1969. 65 pages.
TWI 3-A9. Measurement of time of travel and dispersion in streams by dye tracing, by E.P. Hubbard, F.A. Kilpatrick, L.A. Martens, and J.F. Wilson, Jr. 1982. 44 pages.
TWI 3-A10. Discharge ratings at gaging stations, by E.J. Kennedy. 1984. 59 pages.
TWI 3-A11. Measurement of discharge by moving-boat method, by G.F. Smoot and C.C. Novak. 1969. 22 pages.
TWI 3-A12. Fluorometric procedures for dye tracing, Revised, by James F. Wilson, Jr., Ernest D. Cobb, and Frederick A. Kilpatrick. 1986. 41 pages.
TWI 3-A13. Computation of continuous records of streamflow, by Edward J. Kennedy. 1983. 53 pages.
TWI 3-A14. Use of flumes in measuring discharge, by F.A. Kilpatrick, and V.R. Schneider. 1983. 46 pages.
TWI 3-A15. Computation of water-surface profiles in open channels, by Jacob Davidian. 1984. 48 pages.
TWI 3-A16. Measurement of discharge using tracers, by F.A. Kilpatrick and E.D. Cobb. 1985. 52 pages.
TWI 3-A17. Acoustic velocity meter systems, by Antonius Laenen. 1985. 38 pages.
TWI 3-B1. Aquifer-test design, observation, and data analysis, by R.W. Stallman. 1971. 26 pages.
TWI 3-B2. Introduction to ground-water hydraulics, a programmed text for self-instruction, by G.D. Bennett. 1976. 172 pages. Spanish translation TWI 3-B2 also available.
TWI 3-B3. Type curves for selected problems of flow to wells in confined aquifers, by J.E. Reed. 1980. 106 p.
TWI 3-B5. Definition of boundary and initial conditions in the analysis of saturated ground-water flow systems—an introduction, by O. Lehn Franke, Thomas E. Reilly, and Gordon D. Bennett. 1987. 15 pages.
TWI 3-B6. The principle of superposition and its application in ground-water hydraulics, by Thomas E. Reilly, O. Lehn Franke, and Gordon D. Bennett. 1987. 28 pages.
TWI 3-C1. Fluvial sediment concepts, by H.P. Guy. 1970. 55 pages.
TWI 3-C2. Field methods of measurement of fluvial sediment, by H.P. Guy and V.W. Norman. 1970. 59 pages.
TWI 3-C3. Computation of fluvial-sediment discharge, by George Porterfield. 1972. 66 pages.
TWI 4-A1. Some statistical tools in hydrology, by H.C. Riggs. 1968. 39 pages.
TWI 4-A2. Frequency curves, by H.C. Riggs, 1968. 15 pages.
TWI 4-B1. Low-flow investigations, by H.C. Riggs. 1972. 18 pages.
TWI 4-B2. Storage analyses for water supply, by H.C. Riggs and C.H. Hardison. 1973. 20 pages.
TWI 4-B3. Regional analyses of streamflow characteristics, by H.C. Riggs. 1973. 15 pages.
TWI 4-D1. Computation of rate and volume of stream depletion by wells, by C.T. Jenkins. 1970. 17 pages.
TWI 5-A1. Methods for determination of inorganic substances in water and fluvial sediments, by M.W. Skougstad and others, editors. 1979. 626 pages.
TWI 5-A2. Determination of minor elements in water by emission spectroscopy, by P.R. Barnett and E.C. Mallory, Jr. 1971. 31 pages.

TWI 5-A3. Methods for the determination of organic substances in water and fluvial sediments, edited by R.L. Wershaw, M.J. Fishman, R.R. Grabbe, and L.E. Lowe. 1987. 80 pages. This manual is a revision of "Methods for Analysis of Organic Substances in Water" by Donald F. Goerlitz and Eugene Brown, Book 5, Chapter A3, published in 1972.

TWI 5-A4. Methods for collection and analysis of aquatic biological and microbiological samples, edited by P.E. Greeson, T.A. Ehlke, G.A. Irwin, B.W. Lium, and K.V. Slack. 1977. 332 pages.

TWI 5-A5. Methods for determination of radioactive substances in water and fluvial sediments, by L.L. Thatcher, V.J. Janzer, and K.W. Edwards. 1977. 95 pages.

TWI 5-A6. Quality assurance practices for the chemical and biological analyses of water and fluvial sediments, by L.C. Friedman and D.E. Erdmann. 1982. 181 pages.

TWI 5-C1. Laboratory theory and methods for sediment analysis, by H.P. Guy. 1969. 58 pages.

TWI 6-A1. A modular three-dimensional finite-difference ground-water flow model, by Michael G. McDonald and Arlen W. Harbaugh. 1988. 586 pages.

TWI 7-C1. Finite difference model for aquifer simulation in two dimensions with results of numerical experiments, by P.C. Trescott, G.F. Pinder, and S.P. Larson. 1976. 116 pages.

TWI 7-C2. Computer model of two-dimensional solute transport and dispersion in ground water, by L.F. Konikow and J.D. Bredehoeft. 1978. 90 pages.

TWI 7-C3. A model for simulation of flow in singular and interconnected channels, by R.W. Schaffranek, R.A. Baltzer, and D.E. Goldberg. 1981. 110 pages.

TWI 8-A1. Methods of measuring water levels in deep wells, by M.S. Garber and F.C. Koopman. 1968. 23 pages.

TWI 8-A2. Installation and service manual for U.S. Geological Survey monometers, by J.D. Craig. 1983. 57 pages.

TWI 8-B2. Calibration and maintenance of vertical-axis type current meters, by G.F. Smoot and C.E. Novak. 1968. 15 pages.

CONTENTS

Page

Page

ILLUSTRATIONS

Page

Page

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 are 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 packages can be added to the program without modifying the existing 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 associated with external stresses, such as wells, areal recharge, evapotranspiration, drains, and streams, 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 77 and will run without modification on most computers that have a FORTRAN 77 compiler. For each program module, this report includes a narrative description, a flow chart, a list of variables, and a module listing.

CHAPTER 1

INTRODUCTION

Purpose

Since their inception, the two- and three-dimensional finite-difference models described by Trescott (1975), Trescott and Larson (1976), 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. The basic concepts embodied in those models have been incorporated in the model presented here. The primary objectives in designing a new ground-water flow model were to produce a program that could be readily modified, was simple to use and maintain, could be executed on a variety of computers with minimal changes, and was relatively efficient with respect to computer memory and execution time.

The model program documented in this report uses a modular structure wherein similar program 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 was originally written using FORTRAN 66 (McDonald and Harbaugh, 1984). It has subsequently been modified to use FORTRAN 77. This report documents the FORTRAN 77 version. The program is highly portable; it will run, without modification, on most computers. On some computers, minor modification may be necessary or desirable. A discussion about program portability is contained in Appendix A.

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. Many of the packages represent options which the user may or may not have occasion to use. Each of the packages is described in a separate chapter of this report. Two preliminary chapters

describe topics relating to the overall program; Chapter 2 derives the finite-difference equation that is used in the model and Chapter 3 describes the overall design of the program. Chapter 14 describes utility modules that are used by various packages to perform special tasks. Appendices A-E cover topics relating to the operation of the model.

Chapters 4 through 13 describe individual packages. The description of each package consists of (1) a section entitled "Conceptualization and Implementation," (2) input instructions for the package, and (3) documentation of the individual modules contained in the package. The Conceptualization and Implememention section describes the physical and mathematical concepts used to build the package. For example, in the chapter describing the River Package, an equation is derived which approximates flow through a riverbed, and a discussion is provided to show how that equation can be incorporated into the finite-difference equation. Chapters 12 and 13 describe the solution procedures currently available in the model.

The input instructions in Chapters 4 through 13 are presented in terms of input "items." An item of input may be a single record or a collection of similar records, or it may be an array or a collection of arrays.(In the model described herein, three-dimensional arrays are always read as a collection of two-dimensional arrays, one associated with each model layer.) The input section in each chapter presents a list of the input items associated with the package described in that chapter; the entries in this list are numbered, and generally consist of two lines (sometimes followed by a note or comment). For items which consist of a single record or a group of similar records, the first line in the entry gives the names of the fields comprising the records, while the second line shows the format of those fields, in standard FORTRAN notation. For an input item which consists of an array, the first

line of the entry gives the name of the array, while the second line gives the name of the utility module which reads the array. Further details concerning utility modules are provided in Chapter 14.

For most of the packages, the list of input items is subdivided into two major sections. One of these falls under the heading "FOR EACH SIMULATION" and includes all items for which only one entry is needed in each simulation; the other falls under the heading "FOR EACH STRESS PERIOD", and includes those items for which several entries may be needed in each simulation (for example, pumping rate, which may change with time during the period represented in a simulation). These major sections of the input list are further subdivided by headings which indicate the modules (subroutines) which read the item, or, in the case of an array, which call a utility subroutine to read the array. Input items that are printed entirely in capital letters are used as FORTRAN variables or arrays in the model program; input items which appear in mixed upper and lower case print are terms used in the instructions to describe the input fields or procedures, and do not appear in the model itself as FORTRAN variables. Chapter 4, which describes the Basic Package, includes two lists of input items; one of these describes input which is always required, while the other describes input associated with the optional "output control" section of the Basic Package.

An explanation of input fields is presented following the list of input items in Chapters 4 through 13. This explanation is followed in most cases by a sample input for the package under consideration. In Chapter 4, again, the input items associated with the output control option are treated separately; thus an independent explanation of fields and sample input are

provided for output control.

In each simulation, the user must designate which of the options of the program are to be utilized, and must indicate the file from which the input for each option is to be read. This is done through a one-dimensional array, IUNIT; the entries in this array are the unit numbers associated with the required files by the computer operating system. A location in the IUNIT array is given at the beginning of the input sections in Chapters 5 through 13, and at the beginning of the input discussion for "output control" in Chapter 4. If the option is to be utilized, the user must enter, in the designated IUNIT array location, the unit number of the file or channel through which input for the option is to be read; if the option is not required a zero is entered in this location. Further discussion of the IUNIT array is provided in Chapters 3 and 4.

Following the input section in Chapters 4 through 13, each chapter provides a documentation of the modules making up the associated package. This documentation consists of a list of the modules in the package, followed by detailed descriptions of each of the modules. The detailed description of a module generally 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. If the variable is used only in that module, its range is given as "Module"; if it is used in other modules of the package, but not outside the package, its range is given as "Package"; if it is used in the modules of more than one package, its range is given as "Global."

To summarize the organization of this report, Chapters 2 and 3, and the "Conceptualization and Implementation" section of Chapter 4, provide discussions relevant to the overall design and functioning of the program; the formulation of coefficients representing flow within the aquifer is discussed under "Conceptualization and Implementation" in Chapter 5; Chapters 6 through 11 provide discussions of particular external sources or sinks and their representation in the model; and Chapters 12 and 13 discuss the operation of particular solvers for the systems of finite difference equations generated in the model. Input instructions for each package are provided in the relevant chapter; a discussion of input for utility modules is provided in Chapter 14. The appendices provide a sample problem, abbreviated input instructions, and discussions of certain computer-related topics.

Acknowledgement

The authors wish to extend special thanks to Gordon Bennett. In addition to providing the administrative support for the model development, he provided encouragement and guidance along the way. His critical review of the report greatly improved its clarity.

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

K_{xx}, K_{yy} and K_{zz} are values of hydraulic conductivity along the x, y, and z coordinate axes, which are assumed to be parallel to the major axes of hydraulic conductivity (Lt^{-1});

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).

For a derivation of equation (1) see for example Rushton and Redshaw (1979). In general, S_s, K_{xx}, K_{yy}, and K_{zz} may be functions of space ($S_s = S_s(x,y,z)$, $K_{xx} = K_{xx}(x,y,z)$, etc.) and W may be a function of space and time ($W = W(x,y,z,t)$); equation (1) describes ground-water flow under nonequilibrium conditions in a heterogeneous and anisotropic medium, provided the principal axes of hydraulic conductivity are aligned with the coordinate directions.

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 representation of a ground-water flow system. 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

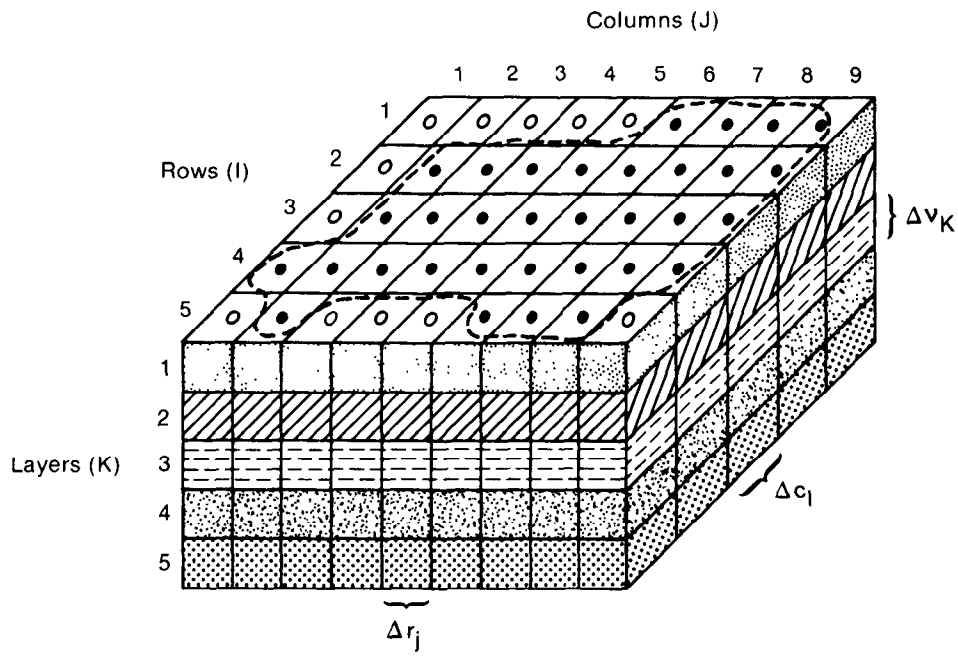
respect to space and time are substituted into equation (1), the equation and its initial and boundary conditions are satisfied. A time-varying head distribution of this nature characterizes the flow system, in that it measures both the energy of flow and the volume of water in storage, and can be used 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 terms calculated from the differences in head 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 times. 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 with a mesh of blocks called cells, the locations of which are described in terms of rows, columns, and layers. An i,j,k indexing system is used. For a system



Explanation

---- Aquifer Boundary

• Active Cell

0 Inactive Cell

Δr_J Dimension of Cell Along the Row Direction Subscript (J) Indicates the Number of the Column

Δc_I Dimension of Cell Along the Column Direction. Subscript (I) Indicates the Number of the Row

Δv_K Dimension of the Cell Along the Vertical Direction. Subscript (K) Indicates the Number of the Layer

Figure 1.—A discretized hypothetical aquifer 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. In formulating the equations of the model, an assumption was made that layers would generally correspond to horizontal geohydrologic units or intervals. Thus in terms of Cartesian coordinates, the k index denotes changes along the vertical, z; because the convention followed in this model is to number layers from the top down, an increment in the k index corresponds to a decrease in elevation. Similarly rows would be considered parallel to the x axis, so that increments in the row index, i, would correspond to decreases in y; and columns would be considered parallel to the y axis, so that increments in the column index, j, would correspond to increases in x. These conventions were followed in constructing figure 1; however, applications of the model requires only that rows and columns fall along consistent orthogonal directions within the layers, and does not require the designation of x, y, or z coordinate axes.

Following the conventions used in figure 1, the width of cells in the row direction, at a given column, j, is designated Δr_j; the width of cells in the column direction at a given row, i, is designated Δc_i; and the thickness of cells in a given layer, k, is designated Δv_k. Thus a cell with coordinates (i,j,k) = (4,8,3) has a volume of $\Delta r_8 \Delta c_4 \Delta v_3$.

Within each cell there is a point called a "node" at which head is to be calculated. Figure 2 illustrates, in two dimensions, 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 orthogonal. 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 chosen so that the hydraulic properties of the system are, in fact, generally uniform over the extent of a cell. The finite-difference equation developed in the following section holds for either formulation; however, only the block-centered formulation is presently used in the model.

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 also 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

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

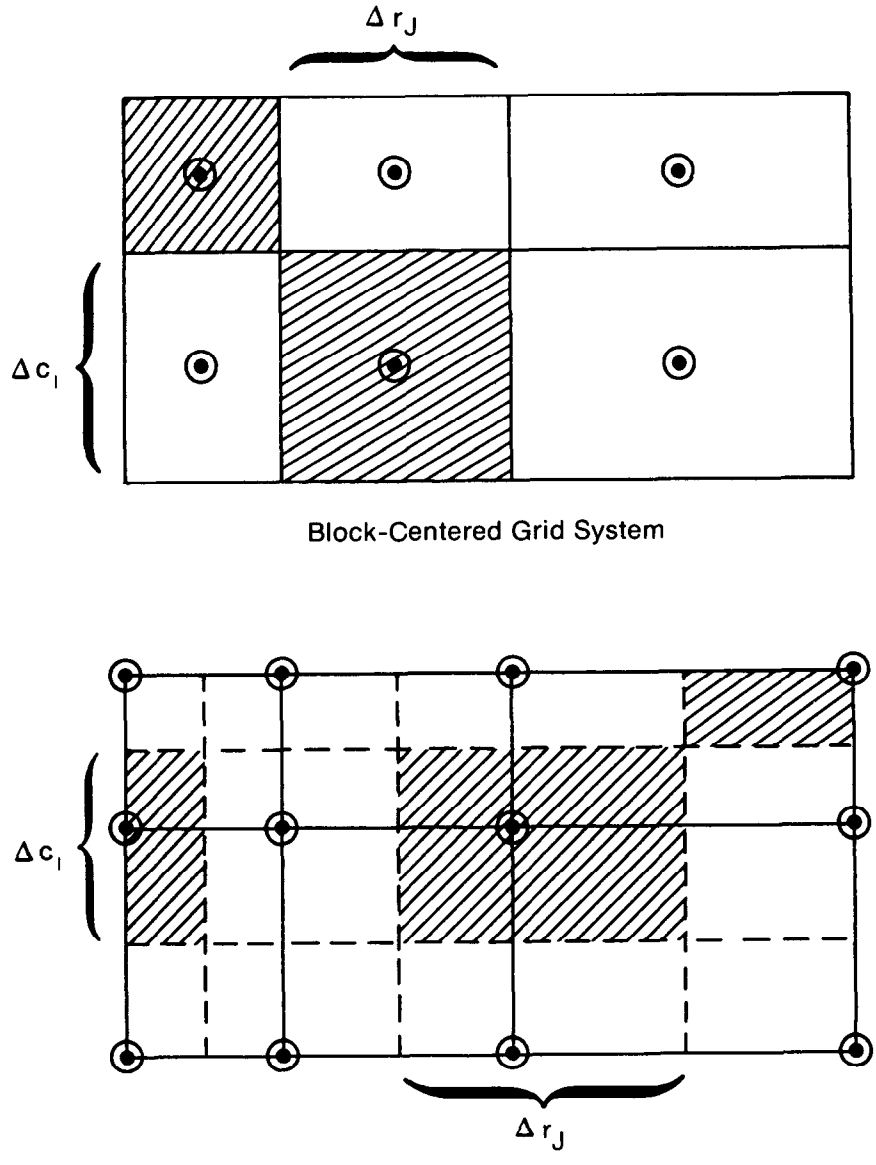


Figure 2.—Grids showing the difference between block-centered and point-centered grids.

where

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

SS has been introduced as the notation for specific storage in the finite-difference formulation; its definition is equivalent to that of S_s in equation (1)--i.e., it is 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. To simplify the following development, flows are considered positive if they are entering cell i,j,k; and the negative sign usually incorporated in Darcy's law has been dropped from all terms. Following these conventions, flow into cell i,j,k in the row direction from cell i,j-1,k (figure 4), is given by Darcy's law as

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

where

$h_{i,j,k}$ is the head at node i,j,k, and $h_{i,j-1,k}$ that at node i,j-1,k;

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

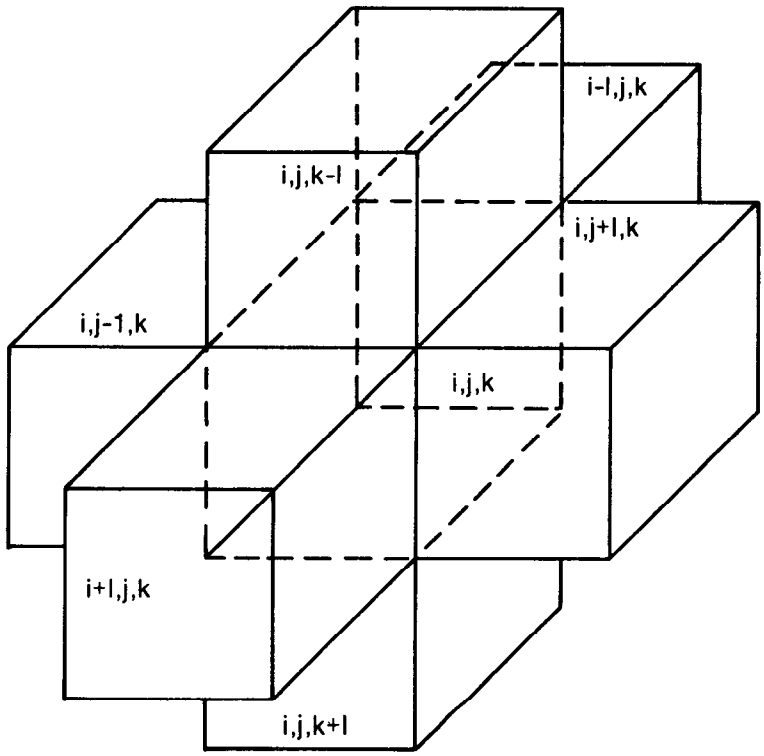


Figure 3.—Cell i,j,k and indices for the six adjacent cells.

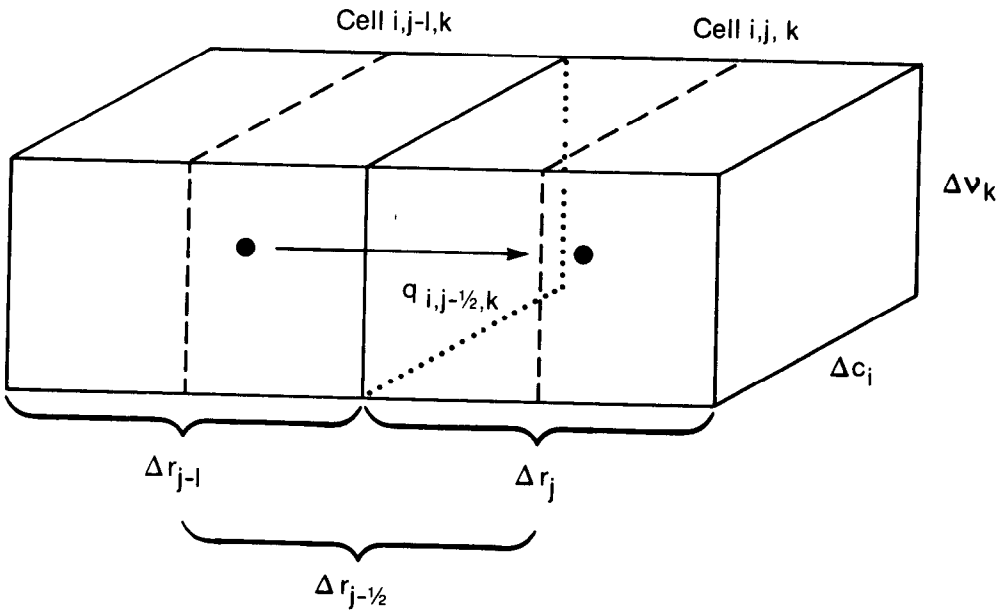


Figure 4.—Flow into cell i,j,k from cell i,j-l,k.

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

$\Delta c_i \Delta v_k$ is the area of the cell faces normal to the row direction; and

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

Although the discussion is phrased in terms of flow into the central cell, it can be misleading to associate the subscript j-1/2 of equation (3) with a specific point between the nodes. Rather, the term $KR_{i,j-1/2,k}$ of equation (3) is the effective hydraulic conductivity for the entire region between the nodes, normally calculated as a harmonic mean in the sense described by, for example, Collins (1961). If this is done, equation (3) gives the exact flow, for a one-dimensional steady-state case, through a block of aquifer extending from node i,j-1,k to node i,j,k and having a cross sectional area $\Delta c_i \Delta v_k$.

Similar expressions can be written approximating the flow into 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 into the block through the forward face is

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

and flow into the block through the rear face is

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

For the vertical direction, inflow through the bottom face is

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

while inflow through the upper face is given by

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

Each of equations (3)-(8) expresses inflow through a face of cell i,j,k in terms of heads, grid dimensions, and hydraulic conductivity. The notation can be simplified by combining grid dimensions and hydraulic conductivity into a single constant, the "hydraulic conductance" or, more simply, the "conductance." For example,

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

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 thus 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 conductance from equation (9) into equation (3) yields

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

Similarly, equations (4)-(8) can be rewritten to yield

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

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

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

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

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

where the 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 features or processes external to the aquifer, such as streams, drains, areal recharge, evapotranspiration or wells, additional terms are required. These flows may be dependent on the head in the receiving cell but independent of all other heads in the aquifer, or they may be entirely independent of head in the receiving cell. Flow from outside the aquifer may be represented by the expression

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

where

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

For example, suppose a cell is receiving flow from two sources, recharge from a well and seepage through a riverbed. For the first source (n=1),

since the flow from the well is assumed to be independent of head, $p_{i,j,k,1}$ is zero and $q_{i,j,k,1}$ is the recharge rate for the well. In this case,

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

For the second source (n=2), the assumption is made that the stream-aquifer interconnection can be treated as a simple conductance, so that the seepage is proportional to the head difference between the river stage and the head in cell i,j,k (figure 5); thus we have

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

where $R_{i,j,k}$ is the head in the river (L) and $CRIV_{i,j,k,2}$ is a conductance (L^2t^{-1}) controlling flow from the river into cell i,j,k. For example, in the situation shown schematically in figure 5, CRIV would be given as the product of the vertical hydraulic conductivity of the riverbed material and the area of the streambed as it crosses the cell, divided by the thickness of the streambed material. 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 negative conductance term, $-CRIV_{i,j,k,2}$ corresponds to $p_{i,j,k,2}$ of equation 16, while 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

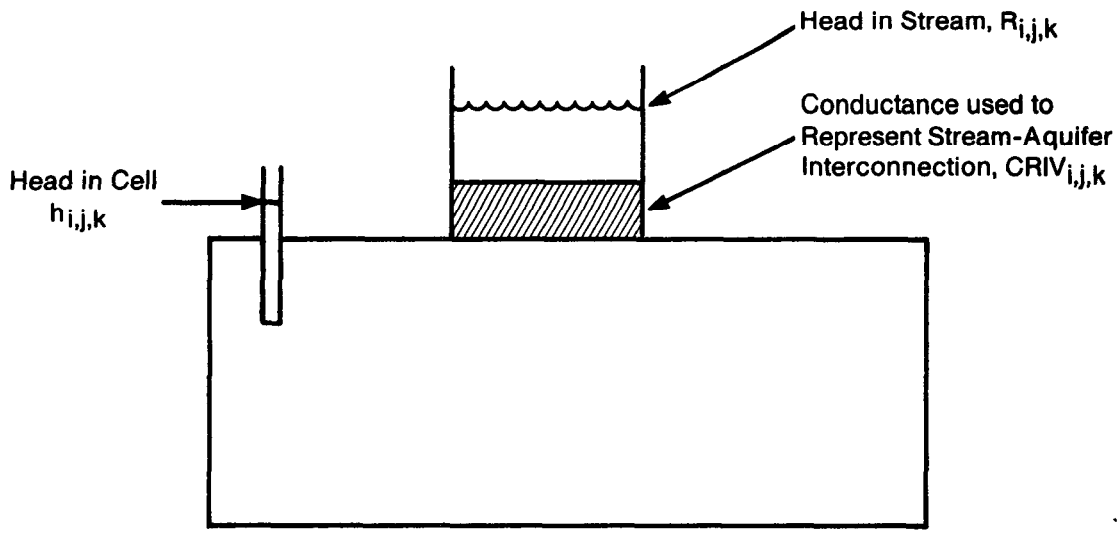


Figure 5.—Conceptual representation of leakage through a riverbed into a cell.

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

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

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

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

Applying the continuity equation (2) to cell i,j,k, taking into account the flows from the six adjacent cells, as well as 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 the derivative of head with respect to time (Lt^{-1});

$SS_{i,j,k}$ represents 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).

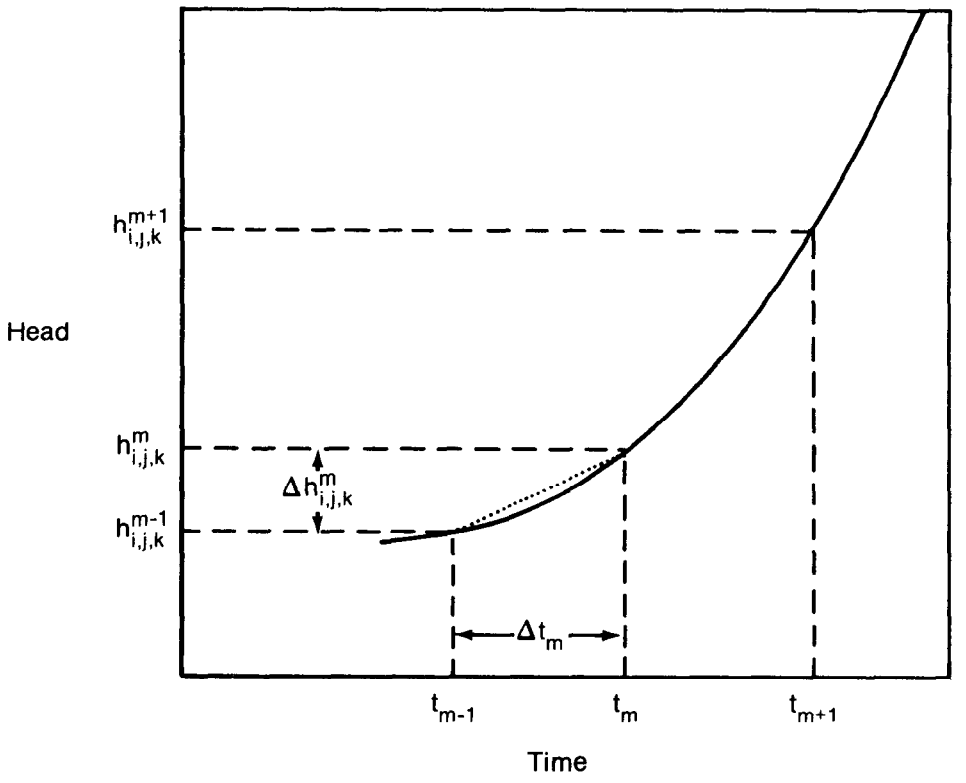
Equations (10) through (15) and (21) may be substituted into equation (22) to give the finite-difference approximation for cell i,j,k as

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

The finite-difference approximation for the time derivative of head, $\frac{\Delta h_{i,j,k}}{\Delta t}$ must next be expressed in terms of specific heads and times. Figure 6 shows a hydrograph of head values at node i,j,k. Two values of time are shown on the horizontal axis: t_m, which is the time at which the flow terms of equation (23) are evaluated; and t_{m-1}, a time which precedes t_m. The head values at node i,j,k associated with these times are designated by superscript as $h^{m}_{i,j,k}$ and $h^{m-1}_{i,j,k}$, respectively. An approximation to the time derivative of head at time t_m is obtained by dividing the head difference $h^{m}_{i,j,k} - h^{m-1}_{i,j,k}$ by the time interval $t_m - t_{m-1}$; that is,

$$\left(\frac{\Delta h_{i,j,k}}{\Delta t}\right)_m \doteq \frac{h^{m}_{i,j,k} - h^{m-1}_{i,j,k}}{t_m - t_{m-1}}$$

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 approximated over a time interval which extends backward in time from t_m, the time at which the flow terms are calculated. 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. These alternatives, however, may cause numerical instability--that is, the growth or propagation of error during the calculation of heads at successive times in a simulation.



Explanation

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

Figure 6.—Hydrograph for cell i,j,k.

In an unstable situation, errors which enter the calculation for any reason at a particular time 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:

$$CR_{i,j-1/2,k}(h^m_{i,j-1,k} - h^m_{i,j,k}) + CR_{i,j+1/2,k}(h^m_{i,j+1,k} - h^m_{i,j,k})$$
$$+ CC_{i-1/2,j,k}(h^m_{i-1,j,k} - h^m_{i,j,k}) + CC_{i+1/2,j,k}(h^m_{i+1,j,k} - h^m_{i,j,k})$$
$$+ CV_{i,j,k-1/2}(h^m_{i,j,k-1} - h^m_{i,j,k}) + CV_{i,j,k+1/2}(h^m_{i,j,k+1} - h^m_{i,j,k})$$
$$+ P_{i,j,k}h^m_{i,j,k} + Q_{i,j,k} = SS_{i,j,k}(\Delta r_j \Delta c_i \Delta v_k)\frac{(h^m_{i,j,k} - h^{m-1}_{i,j,k})}{t_m - t_{m-1}}. \quad (24)$$

Equation (24) is a backward-difference equation which can be used as the basis for a simulation of the partial differential equation of ground water flow, equation (1). Like the term $Q_{i,j,k}$, the coefficients of the various head terms in equation (24) are all known, as is the head at the beginning of the time step, $h^{m-1}_{i,j,k}$. The seven heads at time t_m, the end of the time step, are unknown; that is, they are part of the head distribution to be predicted. Thus equation (24) cannot be solved independently, since it represents a single equation in seven unknowns. However, an equation of this type can be written for each active cell in the mesh; 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 can be solved simultaneously.

The objective of transient simulation is generally to predict head distributions at successive times, given the initial head distribution, the boundary conditions, the hydraulic parameters and the external stresses. The initial-head distribution provides a value of $h^1_{i,j,k}$ at each point in the mesh---that is, it provides the values of head at 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 marks the end of the first time step. In equation (25), therefore, the head superscript m is taken as 2, while the superscript m-1, which appears in only one head term, is taken as 1. The equation therefore becomes

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

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

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

$$+ P_{i,j,k}h^2_{i,j,k} + Q_{i,j,k} = SS_{i,j,k}\frac{(\Delta r_j \Delta c_i \Delta v_k)(h^2_{i,j,k} - h^1_{i,j,k})}{t_2 - t_1} \qquad (25)$$

where again the superscripts 1 and 2 refer to the time at which the heads are taken and should not be interpreted as exponents.

An equation of this form is written for every cell in the mesh in which head is free to vary with time (variable-head cells), and the system of equations is solved simultaneously for the heads at time t_2. When these 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 equations is formulated, where the unknowns are now the heads at time t_3; and this set of equations is solved simultaneously to obtain the head distribution at time 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 step.

Iteration

The model described in this report utilizes iterative methods to obtain the solution to the system of finite-difference equations for each time step. In these methods, the calculation of head values for the end of a given time step is started by arbitrarily assigning a trial value, or estimate, for the head at each node at the end of that step. A procedure of calculation is then initiated which alters these estimated values, producing a new set of head values which are in closer agreement with the system of equations. These new, or interim, head values then take the place of the initially assumed heads, and the procedure of calculation is repeated, producing

a third set of head values. This procedure is repeated successively, at each stage producing a new set of interim heads which more nearly satisfies the system of equations. Each repetition of the calculation is termed an "iteration." Ultimately, as the interim heads approach values which would exactly satisfy the set of equations, the changes produced by succeeding stages of calculation become very small. This behaviour is utilized in determining when to stop iteration, as discussed in a subsequent paragraph.

Thus, during the calculations for a time step, arrays of interim head values are generated in succession, each array containing one interim head value for each active node in the mesh. In figure 7, these arrays are represented as three-dimensional lattices, each identified by an array symbol, $\overline{h}$, bearing two superscripts. The first superscript indicates the time step for which the heads in the array are calculated, while the second indicates the number, or level, of the iteration which produced the head array. Thus $\overline{h}^{m,2}$ represents the array of values computed in the first iteration for the end of step m; $\overline{h}^{m,2}$ would represent the array of values computed in the second iteration; and so on. The head values which were initially assumed for the end of time step m, to begin the process of iteration, appear in the array designated $\overline{h}^{m,o}$. In the example of figure 7, a total of n iterations is required to achieve closure for the heads at the end of time step m; thus the array of final head values for the time step is designated $\overline{h}^{m,n}$. Figure 7 also shows the array of final head values for the end of the preceding time step $\overline{h}^{m-1,n}$ (where again it is assumed that n iterations were required for closure). The head values in this array appear in the storage term of equation (24)--i.e., they are used in the term $h^{m-1}_{i,j,k}$ on the right side of equation (24)--in the calculation of heads for time step m. Because they represent heads for the preceding time step, for which computations have

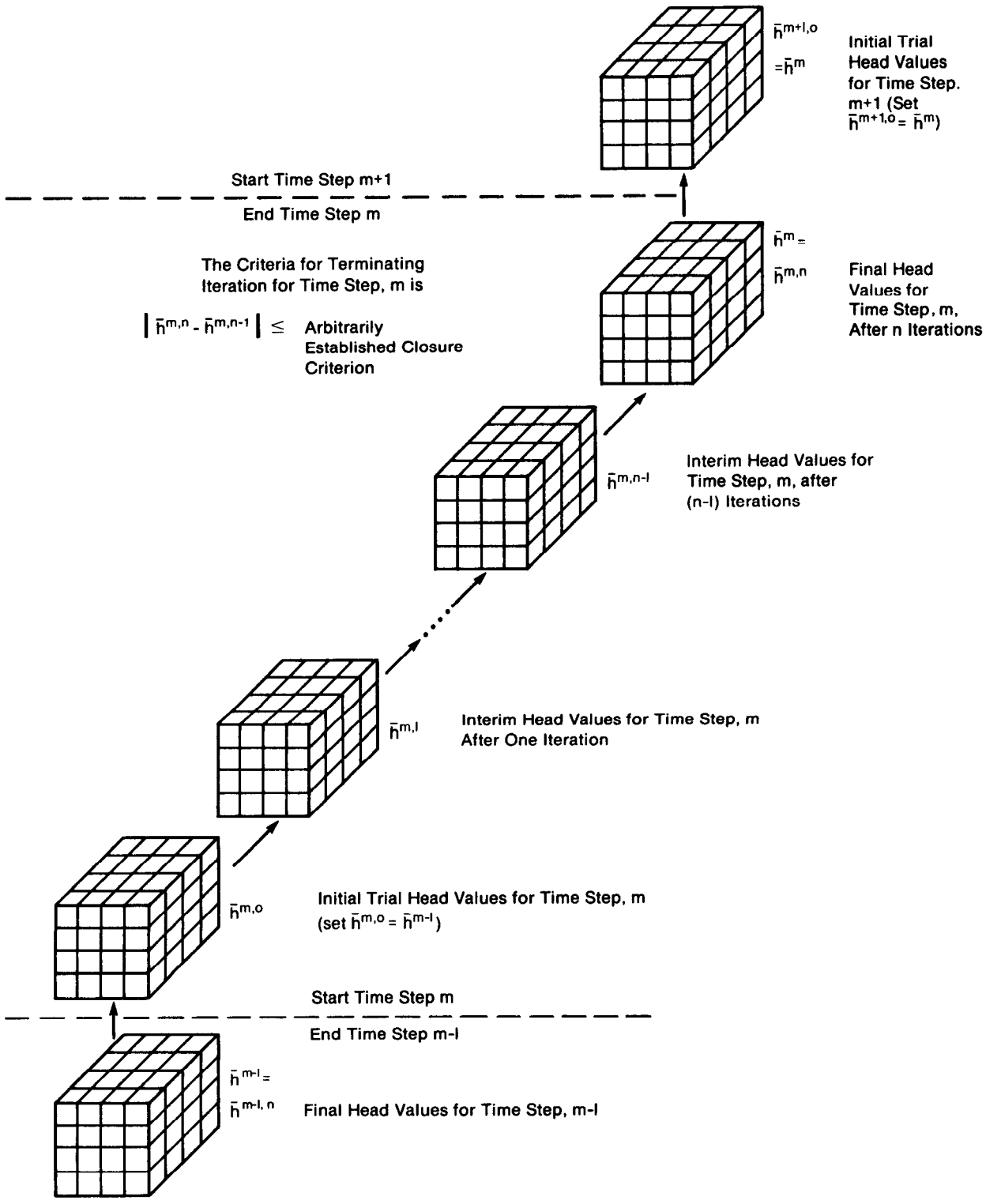


Figure 7.—Iterative calculation of a head distribution.

already been completed, they appear as predetermined constants in the equation for time step m; thus they retain the same value in each iteration of the time step. Similarly, the final values of head for time step m are used as constants in the storage term during calculations for time step m+1.

Ideally, one would like to specify that iteration stop when the calculated heads are suitably close to the exact solution. However, because the actual solution is unknown, an indirect method of specifying when to stop iterating must be used. The method most commonly employed is to specify that the changes in computed heads occuring from one iteration level to the next must be less than a certain quantity, termed the "closure criterion" or "convergence criterion," which is specified by the user. After each iteration, absolute values of computed head change in that iteration are examined for all nodes in the mesh. The largest of these absolute head change values is compared with the closure criterion. If this largest value exceeds the closure criterion, iteration continues; if it is less than the closure criterion, iteration is said to have "closed" or "converged," and the process is terminated for that time step. Normally, this method of determining when to stop iteration is adequate. Note that the closure criterion refers to change in computed head, and that values of head are not themselves necessarily calculated to a level of accuracy comparable to the closure criterion. As a rule of thumb, it is wise to use a value of closure criterion that is an order of magnitude smaller than the level of accuracy desired in the head results.

The program described herein also incorporates a maximum permissible number of iterations per time step. If closure is not achieved within this maximum number of iterations, the iterative process will be terminated and a

corresponding message printed in the output. The closure criterion is designated HCLOSE in the model input, while the maximum number of iterations per time step is designated MXITER.

The initial estimates of head for the end of time step m, in array $\overline{h}^{m,o}$ of figure 7, could be assigned arbitrarily, or they could be chosen according to a number of different conventions. Theoretically, the iterative process would eventually converge to the same result regardless of the choice of initial head values, although the work required would be much greater for some choices than for others. In the model described in this report, the heads computed for the end of each time step are used as the initial trial values of head for the end of the succeeding time step. Thus in figure 7, the array $\overline{h}^{m-1,n}$ contains the final estimates of head, obtained after n iterations, for the end of time step m-1. When the calculations for step m-1 are complete, these same values of head are transferred to the array $\overline{h}^{m,o}$, and used as the initial estimates, or trial values, for the heads at the end of time step m. Head values for the end of the first time step in the simulation are assumed initially to be equal to the heads specified by the user for the beginning of the simulation.

Discussions of the mathematical basis of various iterative methods may be found in many standard references, including Peaceman (1977), Crichlow (1977) and Remson, Hornberger and Molz (1971). It is suggested that the reader review one of these discussions, both to clarify general concepts and to provide an introduction to such topics as the use of matrix notation, the role of iteration parameters, and the influence of various factors on rate of convergence. In particular, such a review is recommended prior to reading Chapters 12 and 13 of this report.

An 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 several factors, including 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 (equation (1)). The discrepancy between the head, $h^m_{i,j,k}$, 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, this error tends to become 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.

Formulation of Equations for Solution

The model described in this report presently incorporates two different options for iterative solution of the set of finite-difference equations, and is organized so that alternative schemes of solution may be added without disruption of the program structure. Whatever scheme of solution is employed, it is convenient to rearrange equation (24) 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. The resulting equation is

$$CV_{i,j,k-1/2}h^{m}_{i,j,k-1} + CC_{i-1/2,j,k}h^{m}_{i-1,j,k} + CR_{i,j-1/2,k}h^{m}_{i,j-1,k}$$

$$+ (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k}$$

$$- CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h^{m}_{i,j,k} + CR_{i,j+1/2,k}h^{m}_{i,j+1,k}$$

$$+ CC_{i+1/2,j,k}h^{m}_{i+1,j,k} + CV_{i,j,k+1/2}h^{m}_{i,j,k+1} = RHS_{i,j,k} \qquad (26)$$

where

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

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

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

The entire system of equations of the form of (26), which includes one equation for each variable-head cell in the mesh, may be written in matrix form as

$$[A]\{h\} = \{q\} \qquad (27)$$

where [A] is a matrix of the coefficients of head, from the left side of equation (26), for all active nodes in the mesh; {h} is a vector of head values at the end of time step m for all nodes in the mesh; and {q} is a vector of the constant terms, RHS, for all nodes of the mesh. The model described in this report assembles the vector {q} and the terms that comprise [A] through a series of subroutines, or "modules". The vector {q} and the terms comprising [A] are then transferred to modules which actually solve the matrix equations for the vector {h} .

Types of Model Cell and Simulation of Boundaries

In practice, it is generally unnecessary to formulate an equation of the form of (24) for every cell in a model mesh, as the status of certain cells is specified in advance in order to simulate the boundary conditions of the problem. In the model described in this report, cells of this type are grouped into two categories--"constant-head" cells and "inactive" (or "no-flow") cells. Constant-head cells are those for which the head is specified in advance, and is held at this specified value through all time steps of the simulation. Inactive or no-flow cells are those for which no flow into or out of the cell is permitted, in any time step of the simulation. The remaining cells of the mesh, termed "variable-head" cells in this report, are characterized by heads which are unspecified and free to vary with time. An equation of the form of (24) must be formulated for each variable-head cell in the mesh, and the resulting system of equations must be solved simultaneously for each time step in the simulation.

Constant-head and no flow cells are used in the model described herein to represent conditions along various hydrologic boundaries. For example, figure 8 shows the map of an aquifer boundary superimposed on an array of cells generated for the model. The aquifer is of irregular shape, whereas the model array is always rectangular in outline; no-flow cells have therefore been used to delete the portion of the array beyond the aquifer boundary. The figure also shows constant-head cells along one section of the boundary; these may be used, for example, where the aquifer is in direct contact with major surface water features. Other boundary conditions, such as areas of constant inflow or areas where inflow varies with head, can be simulated through the use of external source terms or through a combination of no-flow cells and external source terms.

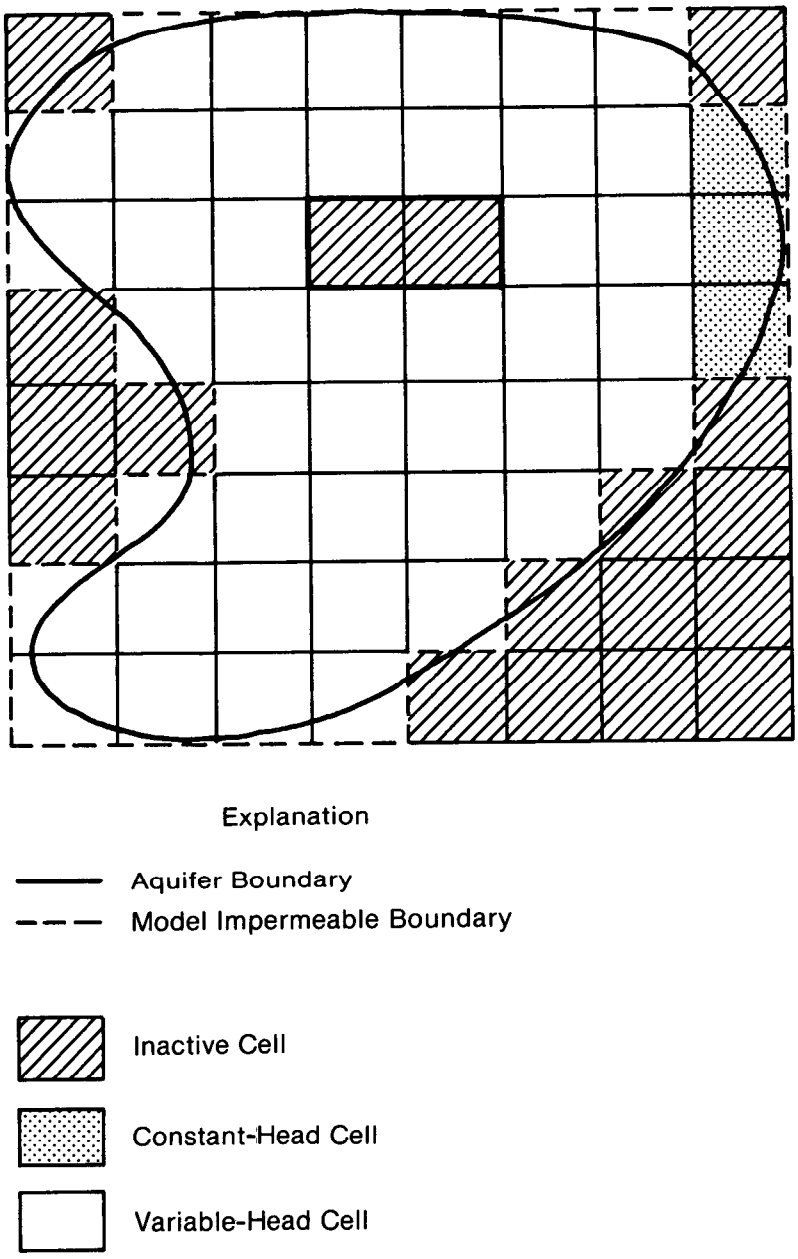
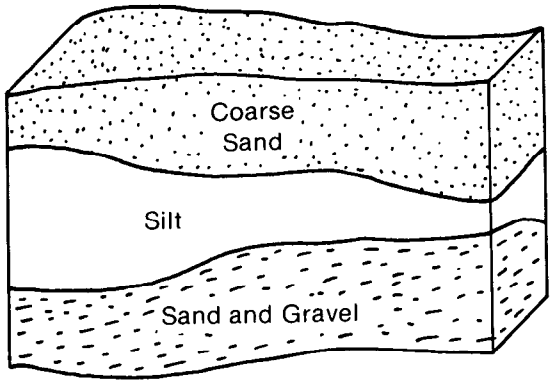


Figure 8.—Discretized aquifer showing boundaries and cell designations.

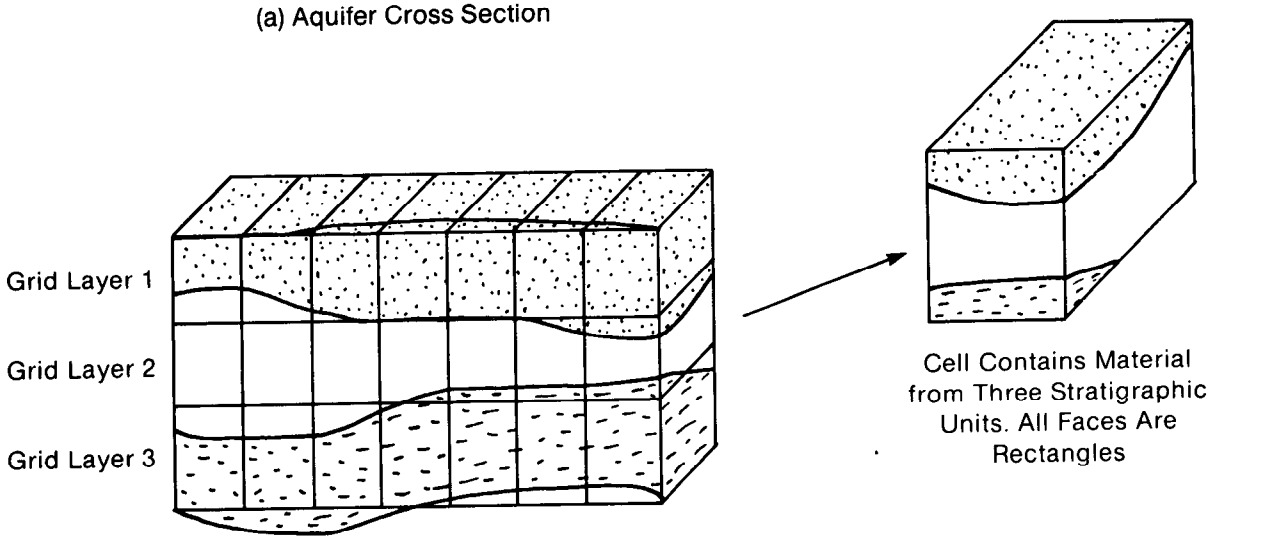
Conceptual Aspects of Vertical Discretization

The model described in this document handles discretization of space in the horizontal direction by reading the number of rows, the number of columns and the width of each row and column (that is, the width of the cells in the direction transverse to the row or column). Discretization of space in the vertical direction is handled in the model by specifying the number of layers to be used, and by specifying hydraulic parameters which contain or embody the layer thickness. This approach is followed in preference to explicit reading of layer thickness in order to accomodate two different ways of viewing vertical discretization.

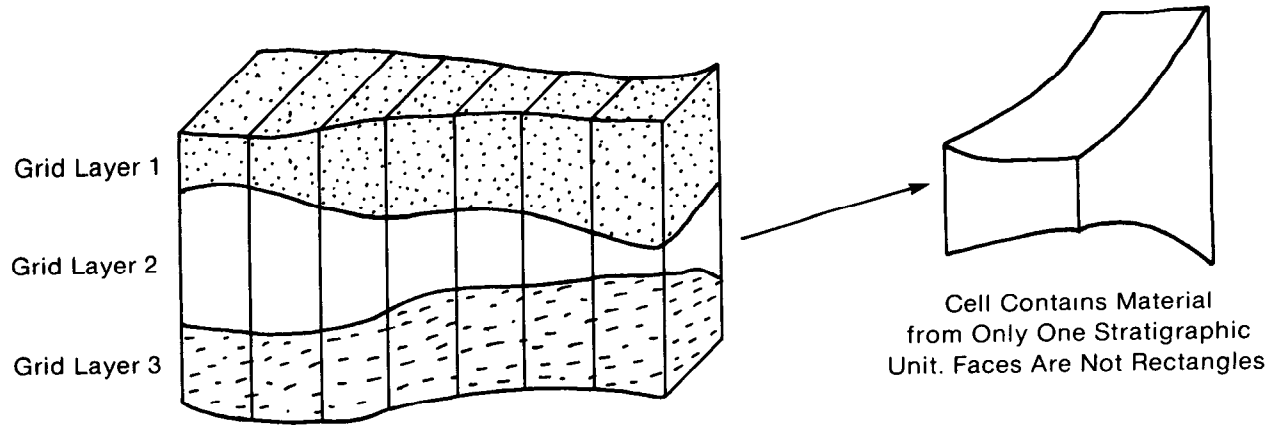
At one extreme, vertical discretization can be visualized simply as an extension of areal discretization--a more or less arbitrary process of dividing the flow system into segments along the vertical, governed in part by the vertical resolution desired in the results. At the opposite extreme, vertical discretization can be viewed as an effort to represent individual aquifers or permeable zones by individual layers of the model. Figure 9-a shows a typical geohydrologic sequence which has been discretized according to both interpretations--in 9-b according to the first viewpoint, and in 9-c according to the second. The first viewpoint leads to rigid superposition of an orthogonal three-dimensional mesh on the geohydrologic system; while there may be a general correspondence between geohydrologic layers and model layers, no attempt is made to make the mesh conform to stratigraphic irregulaties. Under the second viewpoint, model layer thickness is considered variable, to simulate the varying thickness of geohydrologic units; this leads, in effect, to a deformed mesh.



(a) Aquifer Cross Section



(b) Aquifer Cross Section With
Rectilinear Grid Superimposed



(c) Aquifer Cross Section With
Deformed Grid Superimposed

Figure 9.—Schemes of vertical discretization.

Each of these methods of viewing the vertical discretization process has advantages, and each presents difficulties. The model equations are based on the assumption that hydraulic properties are uniform within individual cells, or at least that meaningful average or integrated parameters can be specified for each cell; these conditions are more likely to be met when model layers conform to geohydrologic units as in figure 9-c. Moreover, greater accuracy can be expected if model layers correspond to intervals within which vertical head loss is negligible, and this is also more likely under the configuration of 9-c. On the other hand, the deformed mesh of 9-c fails to conform to many of the assumptions upon which the model equations are based; for example, individual cells may no longer have rectangular faces, and the major axes of hydraulic conductivity may not be aligned with the model axis. Some error is always introduced by these departures from assumed conditions.

In practice many vertical discretization schemes turn out to be a combination of the viewpoints illustrated in figures 9-b and 9-c. For example, even where layer boundaries conform to geohydrologic contacts, it may be necessary to use more than one layer to simulate a single geohydrologic unit, simply to achieve the resolution required in the results. Figure 10 shows a system consisting of two sand units separated by a clay; the units are of uniform thickness, and each could be represented by a single layer without deformation of the mesh. However, flow is neither fully horizontal nor fully vertical in any of the layers; if information on the direction of flow within each unit is required, several layers must be used to represent each unit. Similarly, figure 11 shows a sand-clay system in which pumpage from the sands is sustained partially by vertical flow of water released from storage in the clay. If the objective of analysis is to determine the pattern of storage release in the clay, several model layers would be

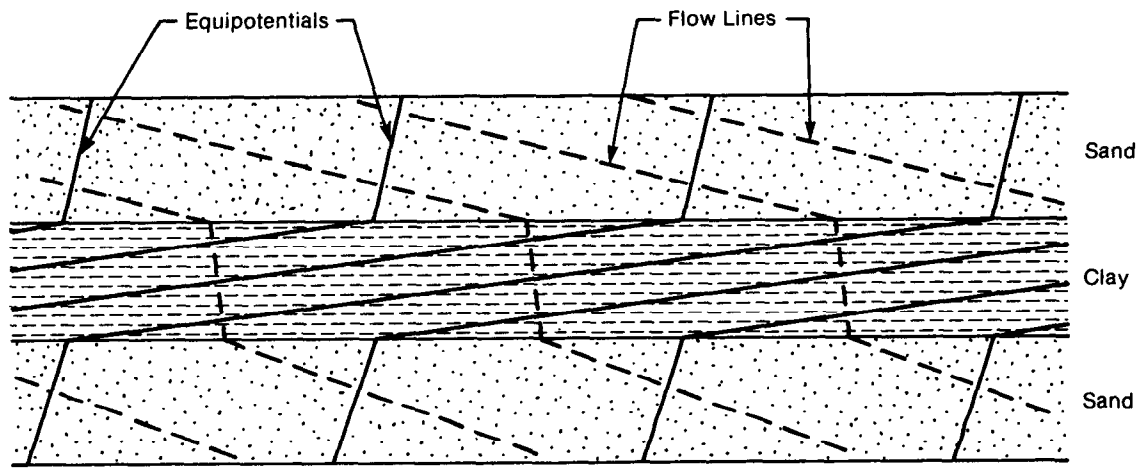


Figure 10.—Possible pattern of flow in a cross section consisting of two high conductivity units separated by a low conductivity unit.

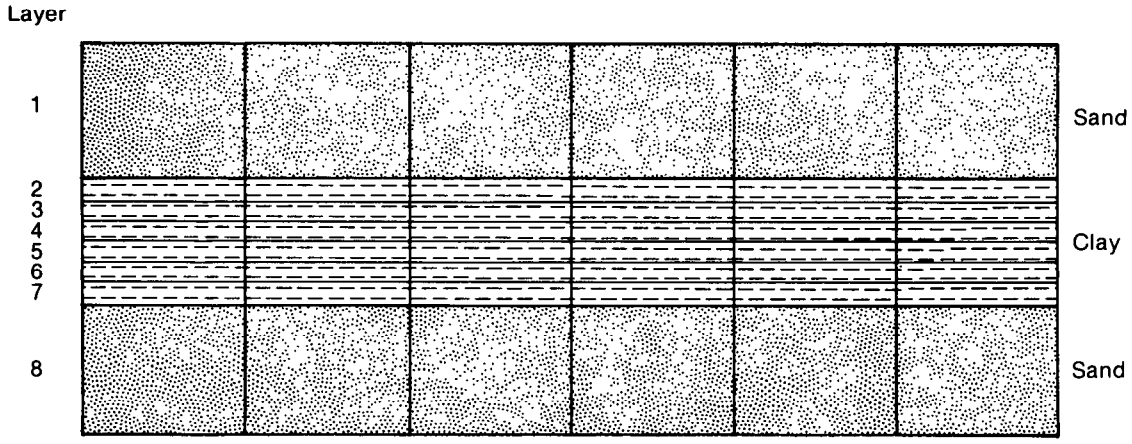


Figure 11.—A cross section in which a low conductivity unit is represented by six model layers.

required to represent that unit, as shown in the figure. On the other hand, figure 12 shows a sand-clay system in which storage release occurs only in the sands, flow in the sand is essentially horizontal, and flow in the clay is essentially vertical. In this case a single model layer may be used to represent each sand, while the clay may be represented simply by the vertical conductance between layers. This approach to vertical discretization has sometimes been termed the "quasi three-dimensional" approach.

The approaches to vertical discretization described above all lead to a set of equations of the form of (26), which must be solved simutaneously at each time step. The differences among these approaches arise in the way the various conductances and storage terms are formulated and, in general, in the number of equations to be solved, the resolution of the results, and the accuracy of the results. The model described in this document is capable of implementing any of these approaches to vertical discretization in that, as noted above, the thickness of individual layers (Δv_k of figure 1 and equation (24)) is never read explicitly by the program; rather, this thickness is embedded in various hydraulic coefficients specified by the user. For example, in confined layers transmissivity, which is the product of hydraulic conductivity and layer thickness, is specified; and storage coefficient, the product of specific storage and layer thickness, is also used. For an unconfined layer, aquifer bottom elevation and hydraulic conductivity are input for each cell. Saturated thickness is calculated as head minus bottom elevation, and transmissivity is then calculated as hydraulic conductivity times saturated thickness. Thus, layer thickness can vary from cell to cell depending on bottom elevation and head. Chapter 5, which describes the Block Centered Flow Package, contains a discussion of the formulation of conductance and storage terms corresponding to the various ways of conceptualizing the vertical discretization.

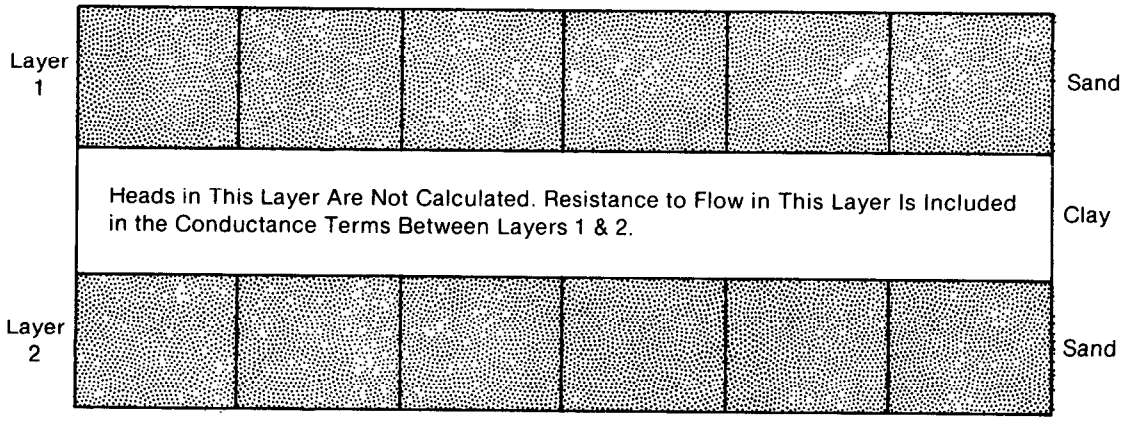


Figure 12.—A cross section in which a low conductivity unit is represented by the conductance between model layers.

CHAPTER 3

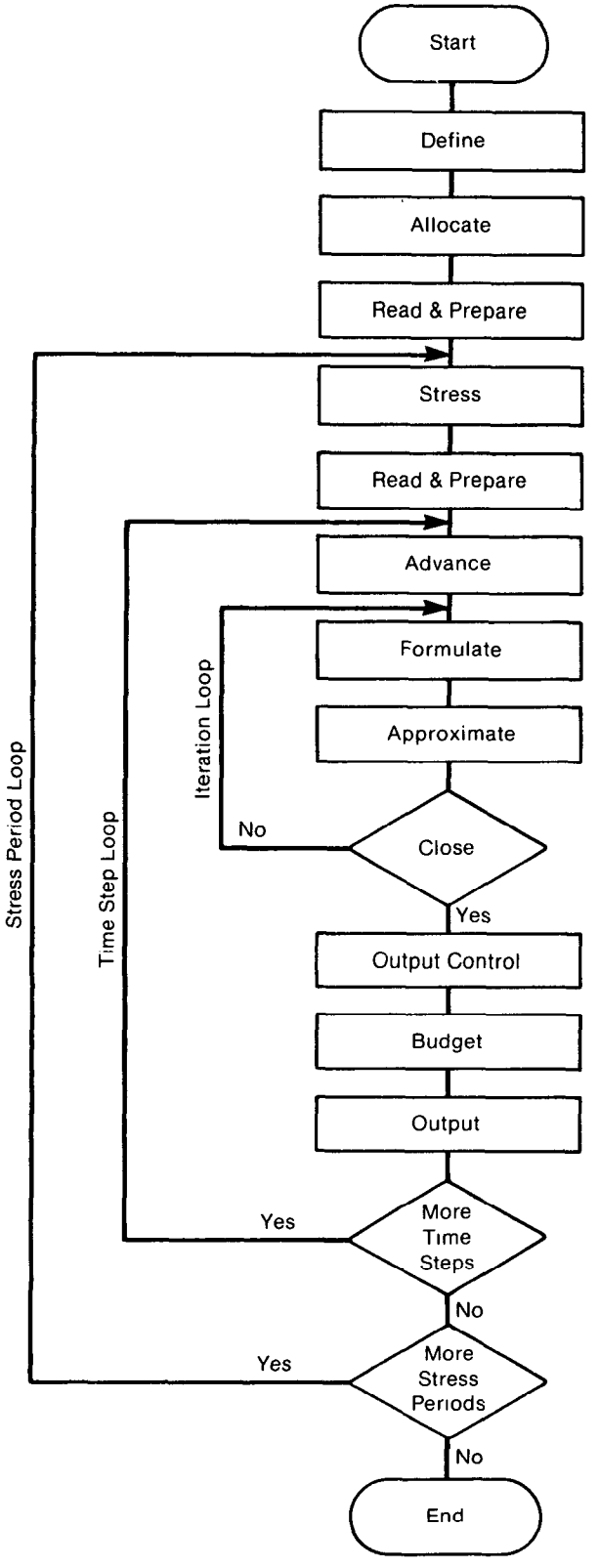
PROGRAM DESIGN

Overall Structure

This chapter describes the overall design of the model 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 can be grouped into "packages" and "procedures".

The functions which must be performed for a typical simulation are shown in figure 13. The period of simulation is divided into a series of "stress periods" within which specified stress parameters are constant. Each stress period, in turn, is divided into a series of time steps. The system of finite-difference equations of the form of equation (27) is formulated and solved to yield the head at each node at the end of each time step. Iterative solution methods are 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 figure 13 is termed a "procedure". For example, prior to entering the stress loop, the program executes three procedures which pertain to the 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 to be used 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



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

ALLOCATE — Allocate space in the computer to store data.

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

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

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

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

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

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

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

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

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

Figure 13.—Overall program structure.

are read. These data may include all or some of the following: boundary conditions, initial heads (starting 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.

Within the stress period loop the first procedure is termed the Stress Procedure. In this procedure the number of time steps (NSTP) in the stress period and certain information to calculate the length of each time step are 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 (figure 13); 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, as explained in a subsequent section. In the Output Procedure, heads, drawdown, and the volumetric budget are printed or recorded.

As shown in the preceding discussion, figure 13 provides a flow chart for the overall program structure, a list of the various procedures, and an indication of the sequence in which those procedures are implemented; it also provides a flow chart for the main program of the model. The work within the procedures--i.e., within the rectangles of figure 13--is performed by individual subroutines, or modules, called by the main program. The main program itself is simply an organized sequence of call statements, most of which are coupled to "IF" tests which determine whether a module is required. Accordingly, the main program does not itself do the work of simulation; it merely calls the various modules in the proper sequence to do that work. Modules which are called directly by the main program are termed "primary" modules; those that are called by other modules are termed "secondary" modules.

Thus the various procedures indicated in figure 13 are implemented through individual modules; and the modules can accordingly be grouped according to the procedure which they help to perform. As noted in Chapter 1, modules can also be grouped by "packages", where a package (for example, the River Package, the Well Package, or the SIP Package) includes those modules required to incorporate a particular hydrologic process or solution algorithm into the simulation. In terms of understanding the operation of the model, these two methods of grouping modules are both useful. The package classification, for example, indicates which modules will be active in a given simulation. (Modules are called by the main program only if they are part of a package which is required in the simulation; and while some packages are required in all simulations, most are needed only when the hydrologic process or solution method embodied in the package is specified by the user.) The procedure classification, on the other hand, defines the

specific function of the module in relation to the functions of other modules of the package. For example, several modules whose function is to allocate space are grouped under the Allocate Procedure; each of these modules allocates the space required for the arrays used in a single package. If few options or features are specified, relatively few packages are involved in the simulation, and the Allocate Procedure is handled by a relatively small number of modules. As the options specified by the user increase, more packages enter the simulation, and more modules are called to complete the space allocation task.

Figure 14 illustrates the classification of modules by procedure and by package in terms of a matrix of primary modules (i.e., modules called by the main program). The horizontal rows in figure 14 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" indicates that the procedure in question is not required in the indicated package. Entries marked with a subscript "S" indicate primary modules which utilize submodules in accomplishing their function; submodules are secondary modules which are utilized only in a single package. Entries marked with the subscript "U" indicate primary modules which utilize utility modules to accomplish their tasks; utility modules are secondary modules which are available to many packages.

Procedures		Flow Component Packages						Solver Packages	
			Stress Packages						
	BAS	BCF	WEL	RCH	RIV	DRN	EVT	GHB	SIP	SOR
Define (DF)	X									
Allocate (AL)	X	X	X	X	X	X	X	X	X	X
Read & Prepare (RP)	X_U	X_{US}							X	X
Stress (ST)	X									
Read & Prepare (RP)			X	X_U	X	X	X_U	X		
Advance (AD)	X									
Formulate (FM)	X	X_S	X	X	X	X	X	X		
Approximate (AP)									X_S	X_S
Output Control (OC)	X									
Budget (BD)		X_{US}	X_U	X_U	X_U	X_U	X_U	X_U		
Output (OT)	X_U									

Figure 14.—Organization of modules by procedures and 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 indicate the procedure. For example, in figure 14, a module is indicated for the Well Package and Allocate Procedure. This module is designated as WEL1AL; the first three letters, WEL, indicate that the module is part of the Well Package; the last two letters, AL, indicate that it performs the Allocate Procedure in that package. Thus this module is one of those that deals with the simulation of specified withdrawal or input, as through wells, and its particular function is to allocate the 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 15 shows the names of the primary modules arranged in the same matrix format that was used in figure 14. As in figure 14, 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

Packages

PROCEDURES	BAS	BCF	WEL	RCH	RIV	DRN	EVT	GHB	SIP	SOR
Define (DF)	BAS1DF									
Allocate (AL)	BAS1AL	BCF1AL	WEL1AL	RCH1AL	RIV1AL	DRN1AL	EVT1AL	GHB1AL	SIP1AL	SOR1AL
Read & Prepare (RP)	$BAS1RP_U$	$BCF1RP_{US}$							SIP1RP	SOR1RP
Stress (ST)	BAS1ST									
Read & Prepare (RP)			WEL1RP	$RCH1RP_U$	RIV1RP	DRN1RP	$EVT1RP_U$	GHB1RP		
Advance (AD)	BAS1AD									
Formulate (FM)	BAS1FM	$BCF1FM_S$	WEL1FM	RCH1FM	RIV1FM	DRN1FM	EVT1FM	GHB1FM		
Approximate (AP)									$SIP1AP_S$	$SOR1AP_S$
Output Control (OC)	BAS1OC									
Budget (BD)		$BCF1BD_{US}$	$WEL1BD_U$	$RCH1BD_U$	$RIV1BD_U$	$DRN1BD_U$	$EVT1BD_U$	$GHB1BD_U$		
Output (OT)	$BAS1OT_U$									

Figure 15.—Primary modules organized by procedure and package.

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.

Table 1 lists the various packages documented in this publication, gives the three-character abbreviation used in the module designation scheme, and provides a brief description of the package operation. Two major categories of package may be recognized--the flow component packages and the solver packages; within the category of flow component packages, a stress package subcategory may be recognized. The flow component packages are those which calculate the coefficients of the finite-difference equation for each cell. This category includes the Block-Centered Flow Package, which formulates the internal flow terms (describing flow between cells and flow to or from storage); and the subcategory of stress packages. Each of the stress packages formulates the coefficients describing a particular external or boundary flow; for example, the River Package calculates the coefficients describing flow between a cell and a surface stream. The solver packages are those which implement algorithms for solution of the systems of finite-difference equations. This documentation describes two packages in this category, one incorporating the Strongly Implicit Procedure of solution, and the other utilizing Slice-Successive Overrelaxation. The only package which does not fit into any of these categories is the Basic Package, which addresses a variety of tasks in support of the entire simulation.

The Block-Centered Flow Package is the only option described in this documentation for the formulation of internal flow terms in the equations. However, alternative packages, for example, utilizing a point centered approach, could certainly be developed and used in place of the Block-Centered Flow Package.

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 porous medium; specifically, flow from cell to cell and flow into storage.		Flow Component Packages
Well	WEL	Adds terms representing flow to wells to the finite-difference equations.	Stress Packages	
Recharge	RCH	Adds terms representing areally distributed recharge to the finite-difference equations.		
River	RIV	Adds terms representing flow to rivers to the finite-difference equations.		
Drain	DRN	Adds terms representing flow to drains to the finite-difference equations.		
Evapotranspiration	EVT	Adds terms representing ET to the finite-difference equations.		
General-Head Boundaries	GHB	Adds terms representing general-head boundaries to the finite-difference equations.		
Strongly Implicit Procedure	SIP	Iteratively solves the system of finite-difference equations using the Strongly Implicit Procedure.		Solver Packages
Slice-Successive Overrelaxation	SOR	Iteratively solves the system of finite-difference equations using Slice-Successive Overrelaxation.		

Similarly, additional solver packages, incorporating different solution algorithms, could be added, as could additional stress packages. Every simulation must include the Basic Package, the Block-Centered Flow Package (or a suitable replacement) and a solver package. Beyond this, the packages to be included in a simulation are at the option of the user, and will depend on the hydrologic processes influencing the problem. The individual modules in the program have been designed in such a way that the packages are totally independent; with the exception of the three required packages noted above, addition or removal of an individual package has no effect on other packages. If an entirely new package is desired, modules can be developed for each of the procedures involved (and the main program modified to call those modules in proper sequence) without affecting other packages of the program.

Figure 16 shows a detailed flow chart of the main program, indicating all of the primary modules together with the tests which determine whether or not each module is to be called. Figure 16 may be studied in conjunction with figures 13 and 15, and table 1, for an appreciation of the overall structure and operation of the model.

The overall design of the model is such that the conductance terms for cell-to-cell flow (CC, CR, and CV of equation (26)) are formulated at the beginning of the simulation, and are reformulated if necessary at each iteration during solution. Reformulation takes place only in unconfined situations, where the conductances depend upon saturated thickness, which may change at each iteration. At present, formulation of conductances is done only by the Block-Centered Flow Package, although again, a replacement package could readily be developed. The lateral conductance terms (CC and CR

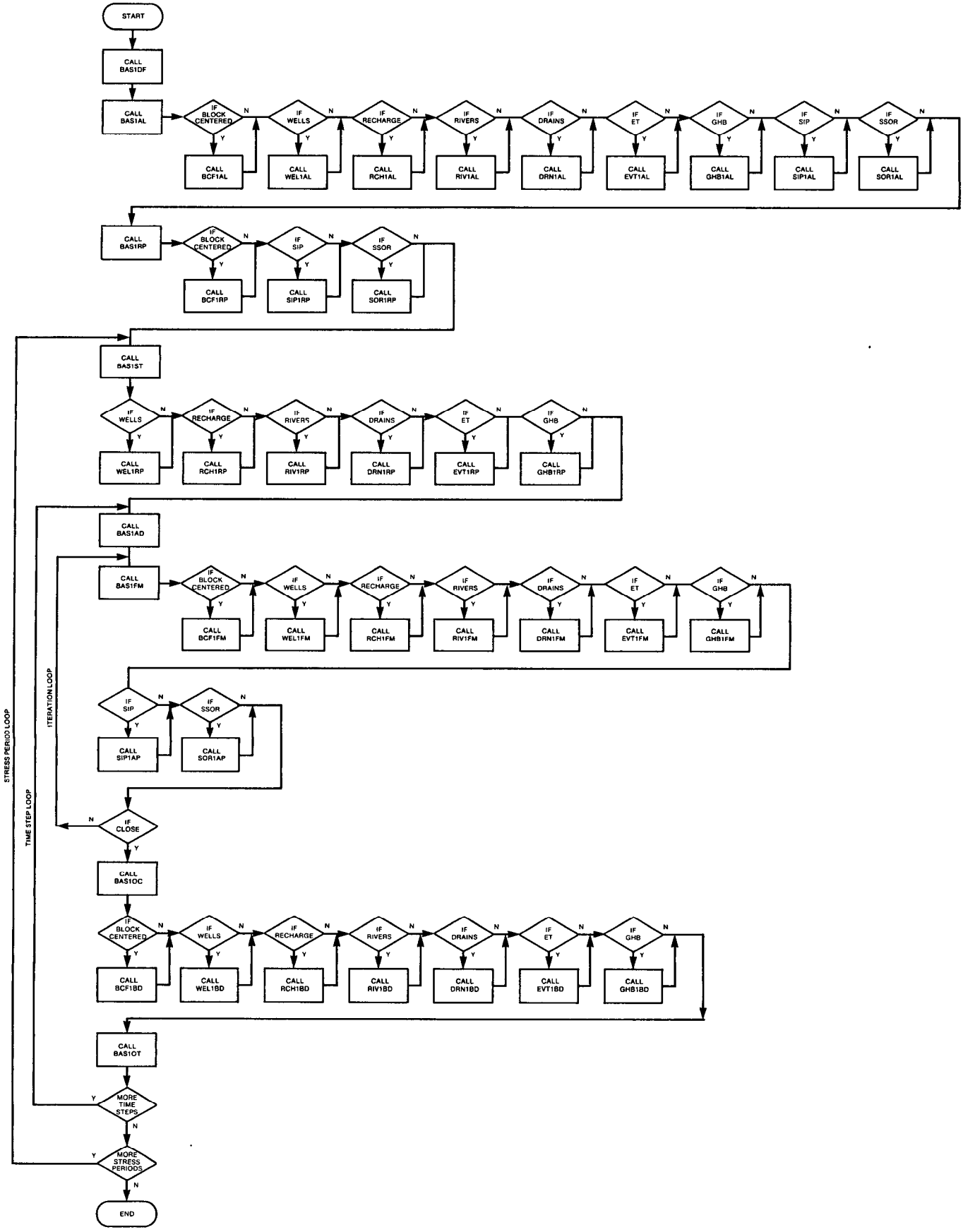


Figure 16.—Overall program structure showing all primary modules.

in equation (26)) are computed as harmonic means for the intervals between nodes, using parameters specified by the user for individual cells; the vertical conductance (CV in equation (26)) is calculated using information which is specified directly for the vertical interval between nodes. The various conductance terms are stored in arrays which are ultimately passed to the solver package, where the matrix equations (equation (27)) are solved.

The coefficient $HCOF_{i,j,k}$ and the term $RHS_{i,j,k}$ of equation (26) are formulated anew at each iteration, for all active nodes in the mesh. This formulation is done progressively, as each package calculates and adds terms for the particular process associated with that package. At the beginning of each iteration, the values of $HCOF_{i,j,k}$ and $RHS_{i,j,k}$ are set to zero throughout the mesh. The Block-Centered Flow Package then adds the term $-SS_{i,j,k}\Delta r_j \Delta c_i \Delta v_k/(t_m - t_{m-1})$ to $HCOF_{i,j,k}$ at each node, and adds the term

$$\frac{-SS_{i,j,k}\Delta r_j \Delta c_i \Delta v_k}{t_m - t_{m-1}} \; h^{m-1}{}_{i,j,k}$$

to $RHS_{i,j,k}$ at each node. For cells that are affected by flow from a stream, given by an expression of the form $P_{si,j,k}\,(h_s - h_{i,j,k})$, where h_s is the (constant) stream head, the River package adds the term $-P_{si,j,k}$ to $HCOF_{i,j,k}$, and adds the constant term $-P_{si,j,k}\,h_s$ to $RHS_{i,j,k}$. This process continues until each package specified by the user has added its contribution to HCOF and RHS at each indicated node of the mesh. The HCOF and RHS arrays are then transferred to the solver package, together with the three conductance arrays (CC,CR and CV), an array containing heads at the beginning of the time step, and the IBOUND array, which identifies constand head, no flow and active nodes. The solver package sums the six conductance terms and the value of *HCOF* at each node to create a single coefficient of $h_{i,j,k}$

(corresponding to the term in brackets in equation (26)), and carries out one iteration of the solution procedure. The various arrays used in the solution procedure are actually stored as segments of a single one-dimensional array, the "X" array.

As noted in Chapter 1, Chapters 4 through 13 of this document discuss the program in terms of individual packages. Each of these chapters contains a detailed description of a particular package, including a listing and discussion of each module included in the package. The remainder of this chapter describes the way boundaries, water budget calculations, space allocation and input-output are handled in the model, and provides a brief description and listing of the main program.

Array Boundaries and Aquifer Boundaries

As noted in Chapter 2, the model may be visualized in terms of a three-dimensional assemblage of cells, each cell associated with a node of the model array. The size of the model array is specified by the user in terms of the number of rows (NROW), number of columns (NCOL) and number of layers (NLAY); these terms define a three-dimensional array of cells in the form of a rectangular box. In formulating the finite-difference equations, cell-to-cell conductance terms are omitted for the exterior of cells on the outer surface of this rectangular array. Thus considering flow along a row, a cell-to-cell conductance term is developed for the interval between column 1 and column 2, but not for the interval to the opposite side of column 1; similarly, a conductance term is developed for the interval between column (NCOL-1) and column (NCOL), but not for the interval beyond column (NCOL). Similar conventions are established in the other two directions, so that in

effect the array is bounded externally by planes across which no cell-to-cell flow occurs. If these boundaries of the model array, which are actually embedded in the program, coincide with impermeable boundaries in the aquifer, they can be relied upon to simulate the no-flow condition along those aquifer boundaries without further intervention by the user. In general, however, the aquifer boundaries will be irregular in form, or will not be of a simple impermeable character. In these cases, the aquifer boundary must be simulated by specifying certain cells within the array as no-flow or constant-head, by using external stress terms, or by using a combination of no-flow cells and external stress terms. This was discussed in Chapter 2, and is further discussed below. It should also be noted that while no cell-to-cell conductance terms are formulated for the interval above the uppermost layer of the model array, flow into this layer from above is frequently represented in the model through external stress terms--for example, terms representing evapotranspiration or stream seepage.

A finite-difference equation of the form of (26) is formulated for each variable-head cell in the mesh. For constant-head cells, no equation is formulated; however, the equation for each variable-head cell adjacent to a constant-head cell contains a term describing flow to and from the constant-head cell. For inactive no-flow cells, no equation is formulated, and no term appears in the equation of any adjacent cell for flow to or from the inactive cell; thus no flow is simulated across the interval between an inactive cell and any adjacent cell.

As pointed out above, the model array as initially generated always has the form of a rectangular box. Where the limits of an aquifer do not coincide with this rectangular shape, inactive cells may be used to delete portions of the array which fall outside the aquifer boundaries; this was

discussed through an example in Chapter 2. As noted in the same example, constant-head cells may be used to represent such features as surface water bodies of constant level which are in full contact with the aquifer. Boundaries which are characterized by a constant rate of flow into or out of the aquifer may be simulated using a no-flow boundary in conjunction with the Well Package, by assigning appropriate withdrawal or recharge rates to nodes just inside the boundary. Boundaries characterized by inflow which varies in proportion to head can be simulated using the General Head Boundary Package or the River Package, where these again are applied to nodes just interior to a no-flow boundary. Use of the River Package would involve specifying artificial streambed conductance and stream-head values at each cell along the boundary, where these values are deliberately chosen in such a way as to duplicate the required head-flow relationships.

Constant-head cells, inactive cells and variable-head cells are distinguished from one another in the model through the IBOUND array, which contains one element for each cell in the mesh. The entry in the IBOUND array for a given cell indicates the type of cell according to the following convention:

```
IBOUND (I,J,K) < 0 . . . .Cell I,J,K is constant head
IBOUND (I,J,K) = 0 . . . .Cell I,J,K is inactive
IBOUND (I,J,K) > 0 . . . .Cell I,J,K is variable head
```

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 values equal to zero are changed to inactive cells by the program.

Volumetric Budget

A summary of all inflows and outflows to a region is generally called a water budget. In this report, the water budget is termed a volumetric

budget because it deals with volumes of water and volumetric flow rates; thus strictly speaking it is not a mass balance, although this term has been used in reference to volumetric budgets in other model reports. The model program calculates a water budget for the overall model as a check on the acceptability of the solution, and in order to provide summarized information on the flow system.

Numerical solution techniques for simultaneous equations do not always result in a correct answer; in particular, iterative solvers may stop iterating before a sufficiently close approximation to the solution is attained. A water budget provides an indication of the overall acceptability of the solution. The system of equations solved by the model actually consists of a flow continuity statement for each model cell. Continuity should also exist for the total flows into and out of the model--that is, the difference between total inflow and total outflow should equal the total change in storage. In the model program, the water budget is calculated independently of the equation solution process, and in this sense may provide independent evidence of a valid solution.

Each flow component package calculates its own contribution to the budget. The total budget as printed in the output does not include internal flows between model cells--only flows into or out of the model as a whole. For example, flow to or from rivers, flow to or from constant head cells, and flow to wells are all included in the overall budget terms. Flow into and out of storage is also considered part of the overall budget inasmuch as accumulation in storage effectively removes water from the flow system, and storage release effectively adds water to the flow--even though neither process, in itself, involves the transfer of water into or out of the ground water regime.

For every time step, the budget module of each flow component package calculates the rate of flow into and out of the system due to the process simulated by the package. The inflows and outflows for each component of flow are stored separately in the VBVL array. Most packages deal with only one such component of flow, but the Block-Centered Flow Package deals with two--flow to constant head cells and flow to storage. In addition to flow, the volumes of water entering and leaving the model during the time step are calculated as the product of flow rate and time step length. Cumulative volumes, from the beginning of the simulation, are then calculated and stored in array VBVL.

Module SBAS1V in the BAS Package uses the inflows, outflows and cumulative volumes in the VBVL array to print the budget at the times requested by the model user. When a budget is printed, the flow rates for the last time step and cumulative volumes from the beginning of simulation are printed for each component of flow. Inflows are printed separately from ouflows; following the convention indicated above, water entering storage is treated as an outflow while water released form storage is treated as an inflow. In addition, total inflow and total outflow are printed, as well as the difference between total inflow and outflow. The difference is then printed as a percent error, calculated using the formula:

$$D = \frac{100(IN-OUT)}{(IN+OUT)/2}$$

where IN is the total inflow to the system, OUT is the total outflow and D is the percent error term. If the model equations are correctly solved, the percent error should be small. In general, flow rates may be taken as an indication of solution validity for the time step to which they apply, while cumulative volumes are an indication of validity for the entire

simulation up to the time of the printout. The budget is printed at the end of each stress period whether requested or not.

There are situations in which it is useful to calculate flow terms for various subregions of the model. To facilitate such calculations, provision has been made to save flow terms for individual cells on disk so they can be used in computations external to the model itself. These individual cell flows are referred to here as "cell-by-cell" flow terms, and are of four general types: (1) cell-by-cell stress flows, or flows into or from an individual cell due to one of the external stresses represented in the model, such as evapotranspiration or recharge; (2) cell-by-cell storage terms, which give the rate of accummulation or depletion of storage in an individual cell; (3) cell-by-cell constant-head flow terms, which give the net flow to or from individual constant-head cells; and (4) internal cell-by-cell flows, which are actually the flows across individual cell faces--that is, between adjacent model cells. These four kinds of cell-by-cell term are further discussed in subsequent paragraphs. To save any of these cell-by-cell terms, two flags in the model input must be set. The input to the Output Control section of the Basic Package includes a flag, ICBCFL, which must be set for each time step for which <u>any</u> cell-by-cell terms are to be saved. In addition, each flow component package includes a flag which is set if the cell-by-cell terms computed by that package are to be saved. Thus if the appropriate flag in the Evapotranspiration Package input is set, cell-by-cell evapotranspiration terms will be saved for each time step for which the ICBCFL flag in the Basic Package input is also set. Three of the four types of cell-by-cell flow terms listed above--storage, constant-head cell and internal flows--are computed in the Block-Centered Flow Package, and thus fall under the control of a single flag, IBCFCB, in the input to that

package. Thus in general all three types are saved on disk if this flag is set, and ICBCFL is also set for the time step. Only flow values are saved in the cell-by-cell disk files; neither water volumes nor cumulative water volumes are included. The flow dimensions are volume per unit time, where volume and time are in the same units used for all model input data. The cell-by-cell flow values are stored in unformatted form to make the most efficient use of disk space; see the narrative for the UBUDSV module for information on how the data are written to disk.

Cell-by-cell stress flows are flow rates into or out of the model, at a particular cell, due to one particular external stress. For example, the cell-by-cell evapotranspiration term for cell i,j,k would give the flow out of the model by evapotranspiration from cell i,j,k. Cell-by-cell stress flows are considered positive if flow is into the cell, and negative if it is out of the cell. A cell-by-cell stress flow value is saved for every model cell, for each stress component for which the cell-by-cell flow is requested. That is, an array the size of the model grid is saved on disk for each requested component of flow. For many of the stress components, flow will be zero at most model cells. For example, when using the River Package, there will be nonzero cell-by-cell budget values only at those cells that are traversed by rivers. Thus the amount of disk space required for cell-by-cell flow terms can be large; a flow value is stored for each model cell even when that value is zero, and terms may be saved at many time steps.

The cell-by-cell storage term gives the net flow to or from storage in a variable-head cell. An array of these terms, one for each cell in the mesh is saved in transient simulations if the appropriate flags are

set. Withdrawal from storage in the cell is considered positive, whereas accumulation in storage is considered negative.

The cell-by-cell constant-head flow term gives the flow into or out of an individual constant-head cell. This term is always associated with the constant-head cell itself, rather than with the surrounding cells which contribute or receive the flow. A constant-head cell may be surrounded by as many as six adjacent variable-head cells. The cell-by-cell calculation provides a single flow value for each constant-head cell, representing the algebraic sum of the flows between that cell and all of the adjacent variable-head cells. A positive value indicates that the net flow is away from the constant-head cell (into the variable-head portion of the mesh); a negative value indicates that the net flow is into the constant-head cell.

The internal cell-by-cell flow values represent flows across the individual faces of a model cell. Three such terms are saved by the Block-Centered Flow Package for each variable-head cell and constant-head cell in the mesh, whenever the appropriate cell-by-cell flags are set. These three terms are flow across the front cell face (between cell i,j,k and i+1,j,k), flow across the right face (between cell i,j,k and i,j+1,k), and flow across the lower face (between cell i,j,k and i,j,k+1). Each of these represents flow between a given cell and a neighboring cell. (Although each cell has six neighbors, only three flow terms are required; flow across the other three sides is accounted for in the calculations of flow for cells adjacent to those sides.) Flows are considered positive if they are in the direction of increasing row number, increasing column number or increasing layer number, and are considered negative if in the opposite directions. These internal cell-by-cell flow values are useful

in calculations of the ground-water flow into various subregions of the model, or in constructing flow vectors.

In theory one could calculate a budget identical to the overall budget by using the cell-by-cell flow terms. This is not always true in practice because in some situations the budgets may be summed differently. The cell-by-cell value at a cell for a given stress or flow component is the net flow for that component, which could possibly include two or more flows of the same type, some negative and some positive. Only the net flow for the cell is saved in the cell-by-cell disk file. In the overall budget calculations as performed in the model, on the other hand, positive and negative flows are assembled separately, so that a negative flow at an individual cell would be added to the outflow term and a positive flow at the same cell would be added to the inflow term. Thus if inflow and outflow terms for the entire model are calculated by summing individual cell-by-cell values, they may differ from the corresponding terms as calculated by the model program in the overall budget. However, the difference between inflow and outflow should be the same for either calculation.

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. The main program contains two statements referring to the length of the X array, both of which appear in the first

part of the program listing. In the listing reproduced with this documentation these statements are COMMON X(30000) and LENX = 30000. The number 30000 in the statements refers to the length of the X array; this number must be increased if the storage requirements of the problem exceed 30000 elements.

Three-Dimensional Subscripts for Model Arrays

The conceptualization and implementation sections of this report designate cell locations by row, column, and layer indices in that order (usually designated as i,j,k), as is customary in scientific literature; however, this order of indices is not the most efficient order for array subscripts in the model program. Many model parameters are declared to be three dimensional arrays and accordingly have row, column and layer subscripts. The order of array subscripts in the FORTRAN language determines how data are stored in computer memory. The design of the program is such that the model array subscripts should be in column, row, and layer order for the most efficient memory access on most computers; this order has been used throughout the program. Typically in the program, J is used for the column subscript, I is used for row, and K for layer, but the order is J,I,K rather than I,J,K. It is important to bear in mind this difference in the subscript ordering when comparing the model program to the conceptualization and implementation sections of the report.

Input Structure

The input structure of the program is designed to permit input to be gathered, as it is needed, from many different files. It is based on an element of the FORTRAN language called the unit number, which identifies the file from which the input is to be read (or to which the output is to be written). The user must provide a link between the name of each input or output file and the corresponding unit number; this is generally done externally to the program, through operating system statements.

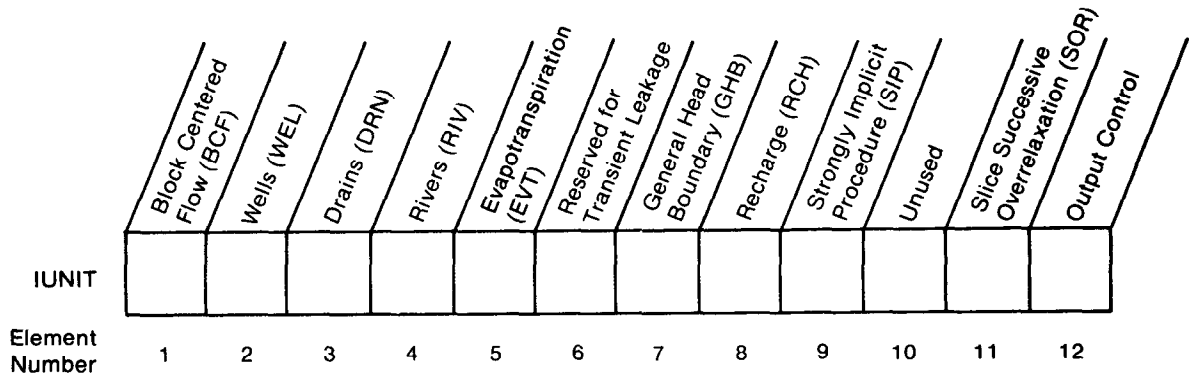
For input purposes, the program may be discussed in terms of "major options"; these are major segments of the program which are utilized only at the user's request. They correspond generally to the individual packages; in fact, all of the existing packages except the Basic Package are considered major options. Output Control, which is not an individual package but rather an optional segment of the Basic Package providing flexibility in program output, is also considered a major option. The balance of the Basic Package is not considered an option since it is always utilized and input for it must always be read. Block-Centered Flow has been treated here as an option, even though it is presently required in all simulations. This has been done to allow for the addition of replacement packages for Block-Centered Flow in the future.

One of the first steps in organizing input data is to specify which of the major options are to be used. This is done using the "IUNIT" array (figure 17) which is read in the Define Procedure by the Basic Package. An option is invoked by inserting an input unit number in the appropriate 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 not to be used, the third element of the IUNIT array (figure 18) is set to zero; if it is to be used, the third element of the array is set to the unit number of the file containing the input data for the package. 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.

As noted above, the Basic (BAS) Package, exclusive of the Output Control option, is used for every simulation; and input data for the Basic Package are always required. Basic Package data (figure 18) 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.

The first element of the IUNIT array must contain the unit number from which data for the Block-Centered Flow (BCF) Package are to be read. At present, because BCF is the only package available for the formulation of cell-to-cell flow terms, a non-zero entry in the first element of IUNIT is always required.

Assignment of Major Options to Elements in the IUNIT Array



Sample IUNIT Input Record

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

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

Figure 17.—Specification of major options using the IUNIT array.

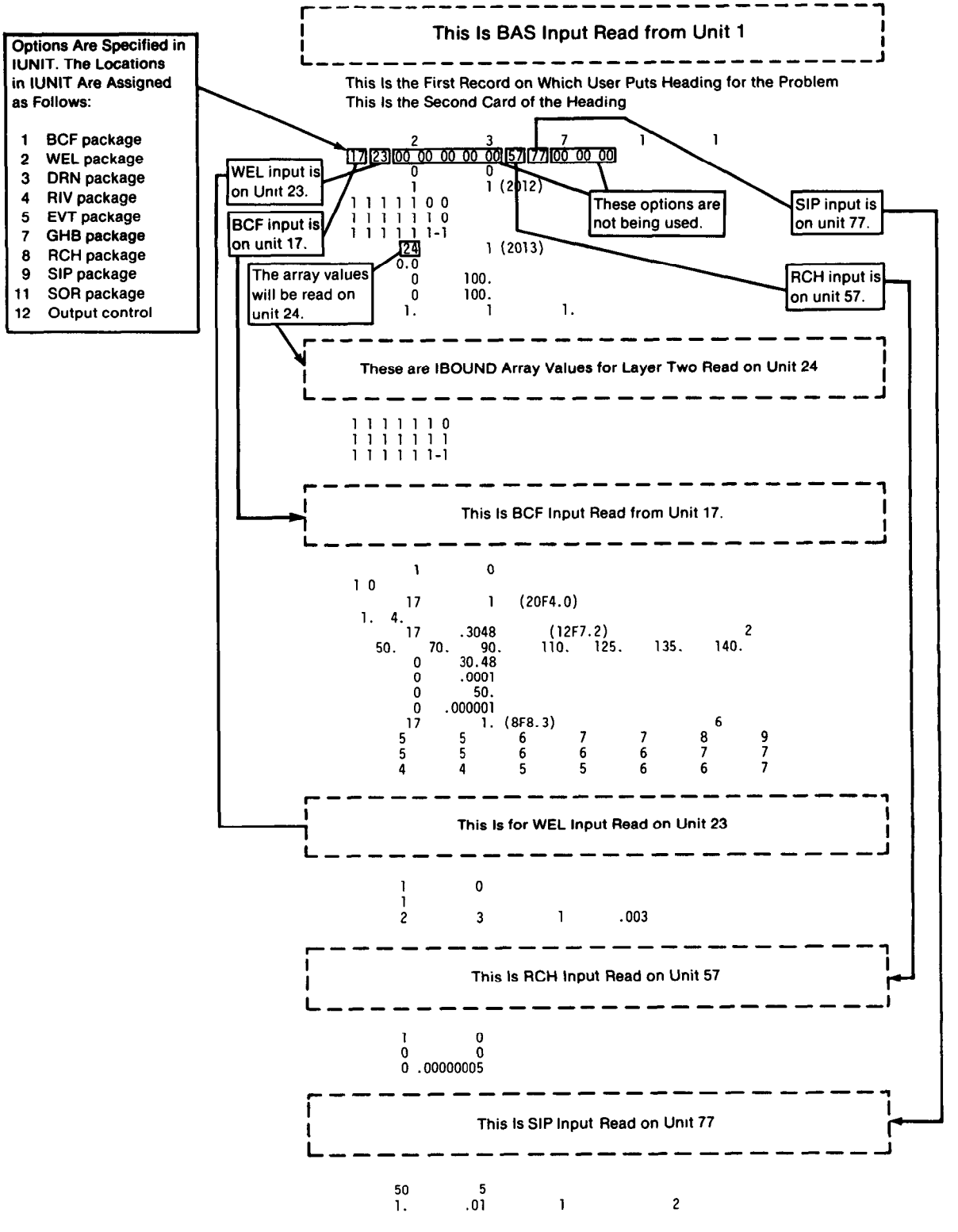


Figure 18.—Sample input data showing role of the IUNIT array.

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 array control record according to 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. Thus there is a great deal of flexibility regarding the organization of the input data for a simulation.

Any consistent length and time units may be used for model 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 in calculations external to the model, or in user-supplied 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. To utilize this option, the user must specify the unit number of the file or channel from which the input data for the Output Control option are to be read. This unit number must be entered as the twelfth element of the IUNIT array (IUNIT 12); the input information is then read, at each time step, from the file identified by this unit number. If a zero is specified as the twelfth element of the IUNIT array, a default output convention is invoked. This default output consists of head values and budget terms printed for 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 (figure 13). 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. The main program calls modules to perform the following tasks, in order (the numbers in the following list correspond to the numbers of the comments in the main program listing).

1. Set the length of the "X" array (LENX) in which all data arrays and lists are stored. Note: LENX should be set equal to the dimension of the X array prior to compilation.

2. Assign the input for the Basic Package to unit 1; assign printed output to unit 6.

3. Define the problem in terms of number of rows, columns, layers, stress periods, and major options to be used.

4. Allocate space in the X array for individual data arrays and lists.

5. If the X array is not big enough for the problem, STOP. (Redimension X and redefine LENX.)

6. Read and prepare information which is constant throughout the simulation.

7. For each stress period:

   (a) Read stress-period timing information.

   (b) Read and prepare information that changes each stress period.

   (c) For each time step:

(1) Calculate the current time-step length and move "new" heads from the preceding time step to the array containing "old" heads of the current time step.

(2) Iteratively formulate and solve the system of equations:

a. Formulate the finite-difference equations.

b. Calculate an approximate solution to the system of equations.

c. If convergence criterion has been met, stop iterating.

(3) Determine the type and amount of output needed for this time step.

(4) Calculate overall budget terms and, if specified, calculate and print or record cell-by-cell flow terms.

(5) Print and/or record heads and/or drawdown. Print the overall volumetric budget and timing summary.

(6) If iteration fails to meet convergence criterion, STOP.

8. END PROGRAM.

```
C     ******************************************************************
C     MAIN CODE FOR MODULAR MODEL --    9/1/87
C          BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C-----VERSION 1638 24JUL1987 MAIN1
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      COMMON X(30000)
      COMMON /FLWCOM/LAYCON(80)
      CHARACTER*4 HEADNG,VBNM
      DIMENSION HEADNG(32),VBNM(4,20),VBVL(4,20),IUNIT(24)
      DOUBLE PRECISION DUMMY
      EQUIVALENCE (DUMMY,X(1))
C     ------------------------------------------------------------------
C
C1------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
      LENX=30000
C
C2------ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
      INBAS=1
      IOUT=6
C
C3------DEFINE PROBLEM__ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
      CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
     1             NODES,INBAS,IOUT,IUNIT)
C
C4------ALLOCATE SPACE IN "X" ARRAY.
      CALL BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1              LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,
     2              INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1AL(ISUM,LENX,LCSC1,LCHY,
     1     LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
     2     NCOL,NROW,NLAY,IOUT,IBCFCB)
      IF(IUNIT(2).GT.0) CALL WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
     1                 IUNIT(2),IOUT,IWELCB)
      IF(IUNIT(3).GT.0) CALL DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
     1                 IUNIT(3),IOUT,IDRNCB)
      IF(IUNIT(8).GT.0) CALL RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     1            NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
      IF(IUNIT(5).GT.0) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
     1            LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
      IF(IUNIT(4).GT.0) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
     1            IUNIT(4),IOUT,IRIVCB)
      IF(IUNIT(7).GT.0) CALL GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,
     1            IUNIT(7),IOUT,IGHBCB)
      IF(IUNIT(9).GT.0) CALL SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,
     1          LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
     2          IUNIT(9),IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,
     1           LCIEQP,MXITER,NCOL,NLAY,NSLICE,MBW,IUNIT(11),IOUT)
C
C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
      IF(ISUM-1.GT.LENX) STOP
C
C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
      CALL BAS1RP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
     1       ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
     2       IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCSC1),
     1          X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELR),
     2           X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
     3           IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT)
```

```
      IF(IUNIT(9).GT.0) CALL SIP1RP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
     1          IUNIT(9),IPCALC,IPRSIP,IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1RP(MXITER,ACCL,HCLOSE,IUNIT(11),
     1         IPRSOR,IOUT)
C
C7------SIMULATE EACH STRESS PERIOD.
      DO 300 KPER=1,NPER
      KKPER=KPER
C
C7A-----READ STRESS PERIOD TIMING INFORMATION.
      CALL BAS1ST(NSTP,DELT,TSMULT,PERTIM,KKPER,INBAS,IOUT)
C
C7B-----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
      IF(IUNIT(2).GT.0) CALL WEL1RP(X(LCWELL),NWELLS,MXWELL,IUNIT(2),
     1           IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1RP(X(LCDRAI),NDRAIN,MXDRN,IUNIT(3),
     1                IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1RP(NRCHOP,X(LCIRCH),X(LCRECH),
     1           X(LCDELR),X(LCDELC),NROW,NCOL,IUNIT(8),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCIEVT),X(LCEVTR),
     1           X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELC),NCOL,NROW,
     1           IUNIT(5),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NRIVER,MXRIVR,IUNIT(4),
     1           IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1RP(X(LCBNDS),NBOUND,MXBND,IUNIT(7),
     1            IOUT)
C
C7C-----SIMULATE EACH TIME STEP.
      DO 200 KSTP=1,NSTP
      KKSTP=KSTP
C
C7C1----CALCULATE TIME STEP LENGTH. SET HOLD=HNEW..
      CALL BAS1AD(DELT,TSMULT,TOTIM,PERTIM,X(LCHNEW),X(LCHOLD),KKSTP,
     1            NCOL,NROW,NLAY)
C
C7C2----ITERATIVELY FORMULATE AND SOLVE THE EQUATIONS.
      DO 100 KITER=1,MXITER
      KKITER=KITER
C
C7C2A---FORMULATE THE FINITE DIFFERENCE EQUATIONS.
      CALL BAS1FM(X(LCHCOF),X(LCRHS),NODES)
      IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
     1          X(LCSC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),X(LCCV),
     2          X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCSC2),
     3          X(LCDELR),X(LCDELC),DELT,ISS,KKITER,KKSTP,KKPER,NCOL,
     4          NROW,NLAY,IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1FM(NWELLS,MXWELL,X(LCRHS),X(LCWELL),
     1           X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(3).GT.0) CALL DRN1FM(NDRAIN,MXDRN,X(LCDRAI),X(LCHNEW),
     1         X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(8).GT.0) CALL RCH1FM(NRCHOP,X(LCIRCH),X(LCRECH),
     1           X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(5).GT.0) CALL EVT1FM(NEVTOP,X(LCIEVT),X(LCEVTR),
     1           X(LCEXDP),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
     1           X(LCHNEW),NCOL,NROW,NLAY)
      IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCRIVR),X(LCHNEW),
     1           X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(7).GT.0) CALL GHB1FM(NBOUND,MXBND,X(LCBNDS),X(LCHCOF),
     1           X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
C
```

```
C7C2B---MAKE ONE CUT AT AN APPROXIMATE SOLUTION.
      IF(IUNIT(9).GT.0) CALL SIP1AP(X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),
     1     X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCFL),X(LCGL),X(LCV),
     2     X(LCW),X(LCHDCG),X(LCLRCH),NPARM,KKITER,HCLOSE,ACCL,ICNVG,
     3     KKSTP,KKPER,IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
     4     IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1     X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCA),X(LCRES),X(LCIEQP),
     2     X(LCHDCG),X(LCLRCH),KKITER,HCLOSE,ACCL,ICNVG,KKSTP,KKPER,
     3     IPRSOR,MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C
C7C2C---IF CONVERGENCE CRITERION HAS BEEN MET STOP ITERATING.
      IF(ICNVG.EQ.1) GO TO 110
  100 CONTINUE
      KITER=MXITER
  110 CONTINUE
C
C7C3----DETERMINE WHICH OUTPUT IS NEEDED.
      CALL BAS1OC(NSTP,KKSTP,ICNVG,X(LCIOFL),NLAY,
     1  IBUDFL,ICBCFL,IHDDFL,IUNIT(12),IOUT)
C
C7C4----CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
      MSUM=1
      IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
     1     X(LCIBOU),X(LCHOLD),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
     2     X(LCTOP),X(LCSC2),DELT,ISS,NCOL,NROW,NLAY,KKSTP,KKPER,
     3     IBCFCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,
     1     X(LCWELL),X(LCIBOU),DELT,NCOL,NROW,NLAY,KKSTP,KKPER,IWELCB,
     1     ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,
     1     X(LCDRAI),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2     KKPER,IDRNCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1BD(NRCHOP,X(LCIRCH),X(LCRECH),
     1     X(LCIBOU),NROW,NCOL,NLAY,DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,
     2     IRCHCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1BD(NEVTOP,X(LCIEVT),X(LCEVTR),
     1     X(LCEXDP),X(LCSURF),X(LCIBOU),X(LCHNEW),NCOL,NROW,NLAY,
     2     DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,IEVTCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1BD(NRIVER,MXRIVR,X(LCRIVR),X(LCIBOU),
     1     X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
     2     KKSTP,KKPER,IRIVCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1BD(NBOUND,MXBND,VBNM,VBVL,MSUM,
     1     X(LCBNDS),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2     KKPER,IGHBCB,ICBCFL,X(LCBUFF),IOUT)
C
C7C5---PRINT AND OR SAVE HEADS AND DRAWDOWNS. PRINT OVERALL BUDGET.
      CALL BAS1OT(X(LCHNEW),X(LCSTRT),ISTRT,X(LCBUFF),X(LCIOFL),
     1        MSUM,X(LCIBOU),VBNM,VBVL,KKSTP,KKPER,DELT,
     2        PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     3     IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
C
C7C6----IF ITERATION FAILED TO CONVERGE THEN STOP.
      IF(ICNVG.EQ.0) STOP
  200 CONTINUE
  300 CONTINUE
C
C8------END PROGRAM
      STOP
C
      END
```

CHAPTER 4

BASIC PACKAGE

Conceptualization and Implementation

The Basic Package handles a number of administrative tasks for the model. It reads data on the number of rows, columns, layers, and stress periods, on the major options to be used, and on the location of input data for those options. It allocates space in computer memory for model arrays; it reads data specifying initial and boundary conditions; it reads and implements data establishing the discretization of time; it sets up the starting head arrays for each time step; it calculates an overall water budget; and it controls model output according to user specification.

Selection of Major Options and Designation of Input Files

The selection of major options and the designation of their input unit numbers were discussed in the preceding chapter. The primary role of the Basic Package in these operations is to read the IUNIT array; as noted in Chapter 3, the entries in this array determine (a) whether or not a major option is to be used and (b) the unit number from which data for the option is to be read. Whenever a new major option is added to the program, an element corresponding to that option must be added to the IUNIT array.

The IBOUND Array

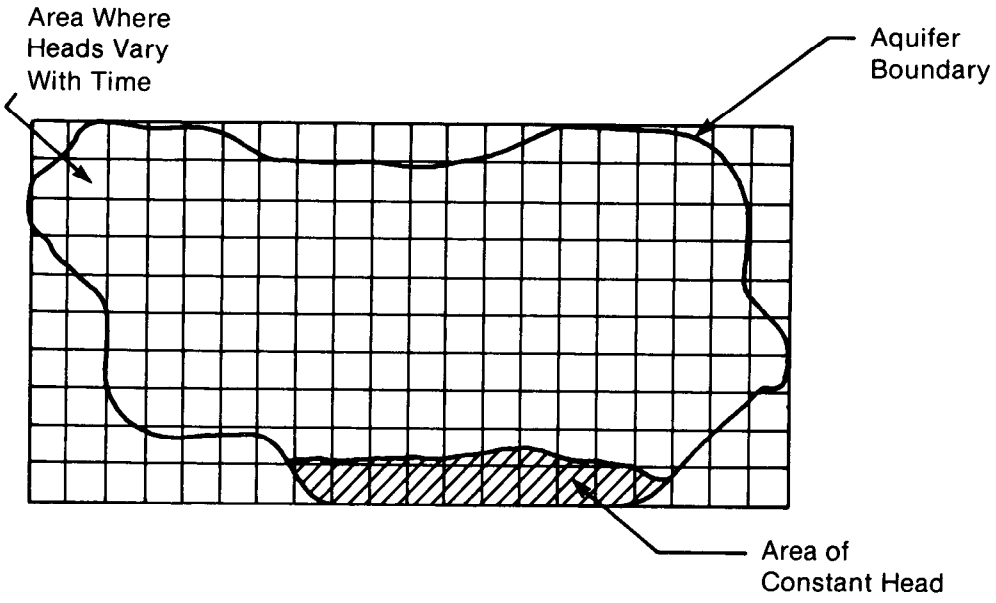
Recall that the finite-difference equation for a cell has the form

$$CR_{i,j-1/2,k}(h^{m}_{i,j-1,k}-h^{m}_{i,j,k}) + CR_{i,j+1/2,k}(h^{m}_{i,j+1,k} - h^{m}_{i,j,k})$$
$$+ CC_{i-1/2,j,k}(h^{m}_{i-1,j,k}-h^{m}_{i,j,k}) + CC_{i+1/2,j,k}(h^{m}_{i+1,j,k} - h^{m}_{i,j,k})$$
$$+ CV_{i,j,k-1/2}(h^{m}_{i,j,k-1}-h^{m}_{i,j,k}) + CV_{i,j,k+1/2}(h^{m}_{i,j,k+1}-h^{m}_{i,j,k})$$
$$+ P_{i,j,k}h^{m}_{i,j,k}+Q_{i,j,k} = SCl_{i,j,k}\,(h^{m}_{i,j,k}-h^{m-1}_{i,j,k})/\Delta t_{m}. \qquad (28)$$

One equation of this form is written for each variable-head cell in the grid. The IBOUND array, which is specified by the user and read by the Basic Package, 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 or inactive cell). The IBOUND array can be modified by other packages if the state of a cell changes. Figure 19 illustrates the distribution of IBOUND code entries for a typical model layer.

Initial Conditions

Because equation (28) is in backward-difference form, a head distribution at the beginning of each time step is required to calculate the head distribution at the end of that time step (figure 20). For each time step after the first, 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. For the first time step, "starting heads" are specified by the user. These specified initial heads are used for head calculation only in the first time step; however, they may also be saved, in the array STRT, and used to



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

IBOUND Codes

< 0 Constant Head
= 0 No Flow
> 0 Variable Head

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

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

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

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

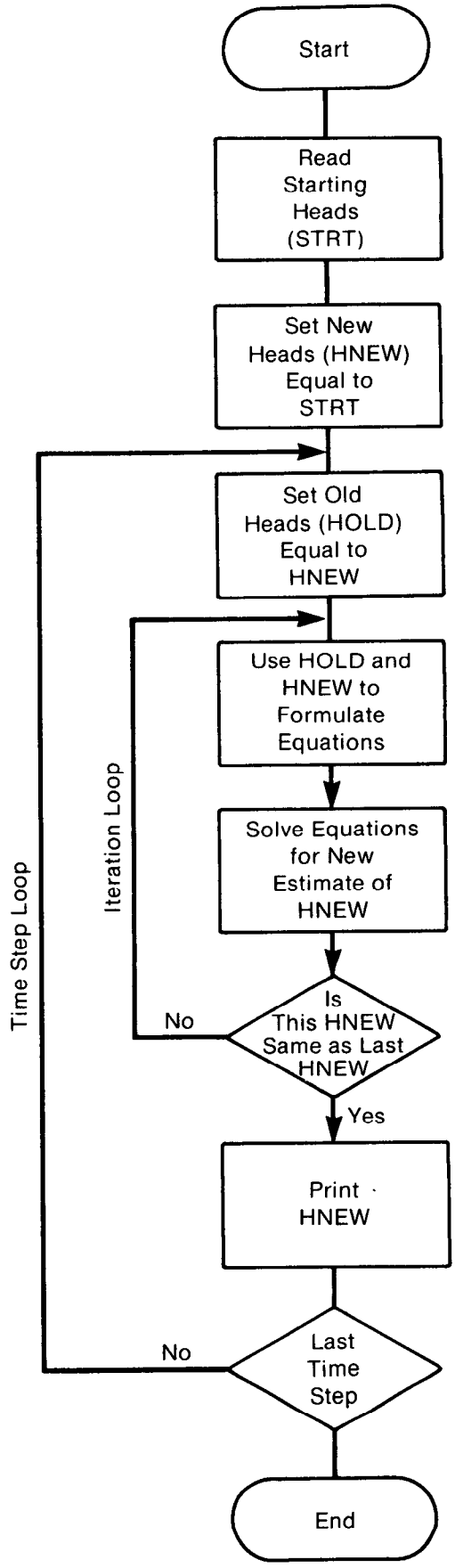


Figure 20.—Flow of head distributions during a simulation.

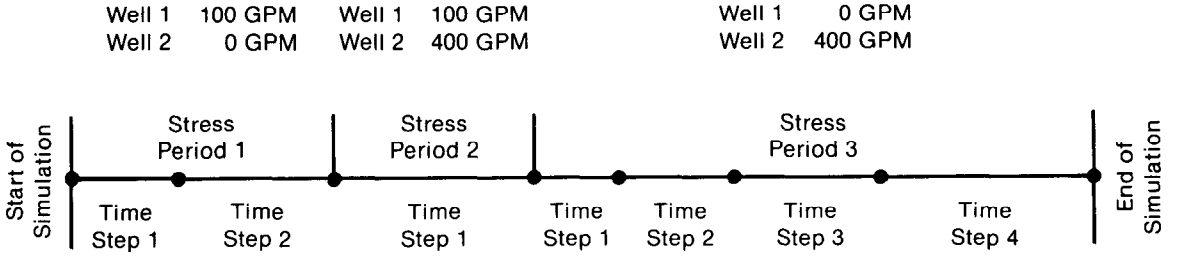
calculate drawdown, the difference between the starting head distribution and some later head distribution.

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 as shown in figure 21. Within each stress period, the time steps form a geometric progression. The user specifies the length of the stress period, the number of time steps into which it is to be divided, and the time step multiplier, or ratio of the length of each time step to that of the preceding time step. Using these terms, the program calculates the length of each time step in the stress period.

Output

The primary output of the program is head distribution. The user may control the frequency at which heads are printed or saved on disk through the "Output Control" option, a major option contained in the Basic Package. Other output items include drawdowns and volumetric budget terms; the Output Control option also provides for storage or printing of these terms. If Output Control is not utilized, a default output option is invoked--the head distribution and the overall volumetric budget are printed at the end of each stress period, and drawdowns are also printed if starting heads were saved. Figure 22 shows an example of a volumetric budget printout for the end of a stress period.



$$Delt(1) = \frac{PERLEN * (1-TSMULT)}{1 - TSMULT ** NSTP}$$

$$Delt(m+1) = TSMULT * Delt(m)$$

Specified by User

- PERLEN ..... Length of Stress Period
- TSMULT ..... Time Step Multiplier
- NSTP ..... Number of Time Steps in Stress Period

Calculated by Program

- Delt(m) ..... Length of Time Step m

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

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

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

      IN:                                                 IN:
      ---                                                 ---
        STORAGE =     .0                                    STORAGE =     .0
  CONSTANT HEAD =     .0                              CONSTANT HEAD =     .0
          WELLS =     .0                                      WELLS =     .0
         DRAINS =     .0                                     DRAINS =     .0
       RECHARGE =     .13608E+08                           RECHARGE =     157.50
       TOTAL IN =     .13608E+08                           TOTAL IN =     157.50
     OUT:                                                OUT:
     ----                                                ----
        STORAGE =     .0                                    STORAGE =     .0
  CONSTANT HEAD =     .43265E+07                      CONSTANT HEAD =     50.075
          WELLS =     .64800E+07                              WELLS =     75.000
         DRAINS =     .28010E+07                             DRAINS =     32.419
       RECHARGE =     .0                                   RECHARGE =     .0
      TOTAL OUT =     .13607E+08                          TOTAL OUT =     157.49
       IN - OUT =     303.00                               IN - OUT =     .34943E-02
PERCENT DISCREPANCY =                 0.00          PERCENT DISCREPANCY =                 0.00
```

Figure 22.--Sample overall volumetric water budget.

Budget Calculations in the Basic Package

The calculation of the volumetric budget is carried out in two parts, the calculation of budget entries and the summation of those entries. As explained in Chapter 3 the entries, which correspond to individual components of flow, are calculated in the flow component packages and stored in the one-dimenisonal array VBVL. The array VBVL is passed to the Basic Package which sums and prints 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)

NOTE: IBOUND and Shead are treated as three-dimensional arrays in the program. However, the input to each of these arrays is handled as a series of two-dimensional arrays, one 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 has been reserved for a transient leakage package, while element 10 has been reserved for an additional solver, both on the assumption that such packages will be added to the model in the future. 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 $\neq$ 0, the arrays BUFF and RHS occupy different space. This option is not needed in the program as documented in this publication. It may be needed for packages yet to be written.

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 (no-flow).

If IBOUND(I,J,K) > 0, cell I,J,K is variable-head.

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 ITEM	EXPLANATION	INPUT RECORDS
1	{HEADNG}	SAMPLE INPUT FOR BASIC PACKAGE
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
	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 -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
	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 -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
	STARTING HEAD VALUES FOR LAYER 1	10. 17. 24. 31. 38. 45. 52. 59. 66. 73. 80. 87. 11. 19. 27. 35. 43. 51. 59. 67. 75. 83. 91. 99. 13. 23. 33. 43. 53. 63. 73. 83. 93. 103. 113. 123. 16. 29. 42. 55. 68. 81. 94. 107. 120. 133. 146. 159. 20. 37. 54. 71. 88. 105. 122. 139. 156. 173. 190. 207. 20. 37. 54. 71. 88. 105. 122. 139. 156. 173. 190. 207. 16. 29. 42. 55. 68. 81. 94. 107. 120. 133. 146. 159. 13. 23. 33. 43. 53. 63. 73. 83. 93. 103. 113. 123. 11. 19. 27. 35. 43. 51. 59. 67. 75. 83. 91. 99. 10. 17. 24. 31. 38. 45. 52. 59. 66. 73. 80. 87.
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

BAS10C

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

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

Explanation of Fields Used in Input Instructions

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

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

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

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

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

INCODE--is the head/drawdown 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.

SAMPLE INPUT TO THE OUTPUT CONTROL OPTION

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

Module Documentation for the Basic Package

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

Primary Modules

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

Submodules

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

Narrative for Module BAS1DF

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

1. Print the name of the program.

2. Read and print a heading.

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

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

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

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

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

8. RETURN.

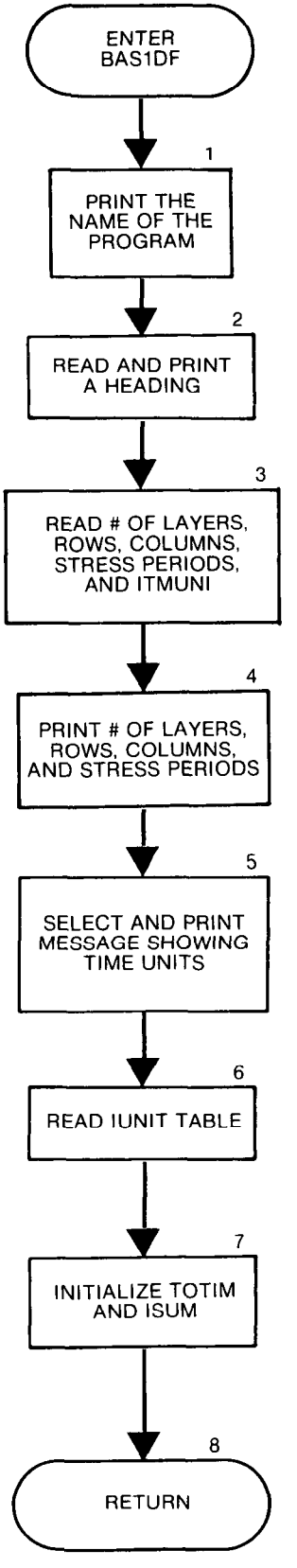
Flow Chart for Module BAS1DF

ITMUNI is a code which indicates units of time used in the input data. This code is only used to print a table showing elapsed time in seconds, minutes, hours, days, and years. It is not used in formulating or solving the finite-difference equation.

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

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

ISUM is a location counter for the first unallocated space in the X array. It is incremented by each module in the Allocate Procedure.



```
      SUBROUTINE BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,
     1       NLAY,NODES,INBAS,IOUT,IUNIT)
C
C-----VERSION 1513 12MAY1987 BAS1DF
C     ******************************************************************
C     DEFINE KEY MODEL PARAMETERS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 HEADNG
      DIMENSION HEADNG(32),IUNIT(24)
C     ------------------------------------------------------------------
C
C1------PRINT THE NAME OF THE PROGRAM.
      WRITE(IOUT,1)
    1 FORMAT(1H1,20X,'U.S. GEOLOGICAL SURVEY MODULAR',
     1       ' FINITE-DIFFERENCE GROUND-WATER MODEL')
C
C2------READ AND PRINT A HEADING.
      READ(INBAS,2) HEADNG
    2 FORMAT(20A4)
      WRITE(IOUT,3) HEADNG
    3 FORMAT(1H0,32A4)
C
C3------READ NUMBER OF LAYERS,ROWS,COLUMNS,STRESS PERIODS AND
C3------UNITS OF TIME CODE.
      READ(INBAS,4) NLAY,NROW,NCOL,NPER,ITMUNI
    4 FORMAT(8I10)
C
C4------PRINT # OF LAYERS, ROWS, COLUMNS AND STRESS PERIODS.
      WRITE(IOUT,5) NLAY,NROW,NCOL
    5 FORMAT(1X,I4,' LAYERS',I10,' ROWS',I10,' COLUMNS')
      WRITE(IOUT,6) NPER
    6 FORMAT(1X,I3,' STRESS PERIOD(S) IN SIMULATION')
C
C5------SELECT AND PRINT A MESSAGE SHOWING TIME UNITS.
      IF(ITMUNI.LT.0 .OR. ITMUNI.GT.5) ITMUNI=0
      GO TO (10,20,30,40,50),ITMUNI
      WRITE(IOUT,9)
    9 FORMAT(1X,'MODEL TIME UNITS ARE UNDEFINED')
      GO TO 100
   10 WRITE(IOUT,11)
   11 FORMAT(1X,'MODEL TIME UNIT IS SECONDS')
      GO TO 100
   20 WRITE(IOUT,21)
   21 FORMAT(1X,'MODEL TIME UNIT IS MINUTES')
      GO TO 100
   30 WRITE(IOUT,31)
   31 FORMAT(1X,'MODEL TIME UNIT IS HOURS')
      GO TO 100
   40 WRITE(IOUT,41)
   41 FORMAT(1X,'MODEL TIME UNIT IS DAYS')
      GO TO 100
   50 WRITE(IOUT,51)
   51 FORMAT(1X,'MODEL TIME UNIT IS YEARS')
C
C6------READ & PRINT INPUT UNIT NUMBERS (IUNIT) FOR MAJOR OPTIONS.
  100 READ(INBAS,101) IUNIT
  101 FORMAT(24I3)
      WRITE(IOUT,102) (I,I=1,24),IUNIT
  102 FORMAT(1H0,'I/O UNITS:'/1X,'ELEMENT OF IUNIT:',24I3,
     1                        /1X,'        I/O UNIT:',24I3)
C
C7------INITIALIZE TOAL ELAPSED TIME COUNTER STORAGE ARRAY COUNTER
C7------AND CALCULATE NUMBER OF CELLS.
      TOTIM=0.
      ISUM=1
      NODES=NCOL*NROW*NLAY
C
C8------RETURN
      RETURN
      END
```

List of Variables for Module BAS1DF

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

Narrative for Module BAS1AL

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

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

Module BAS1AL performs its functions in the following order:

1. Print a message identifying the package.

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

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

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

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

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

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

8. RETURN.

Flow Chart for Module BAS1AL

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

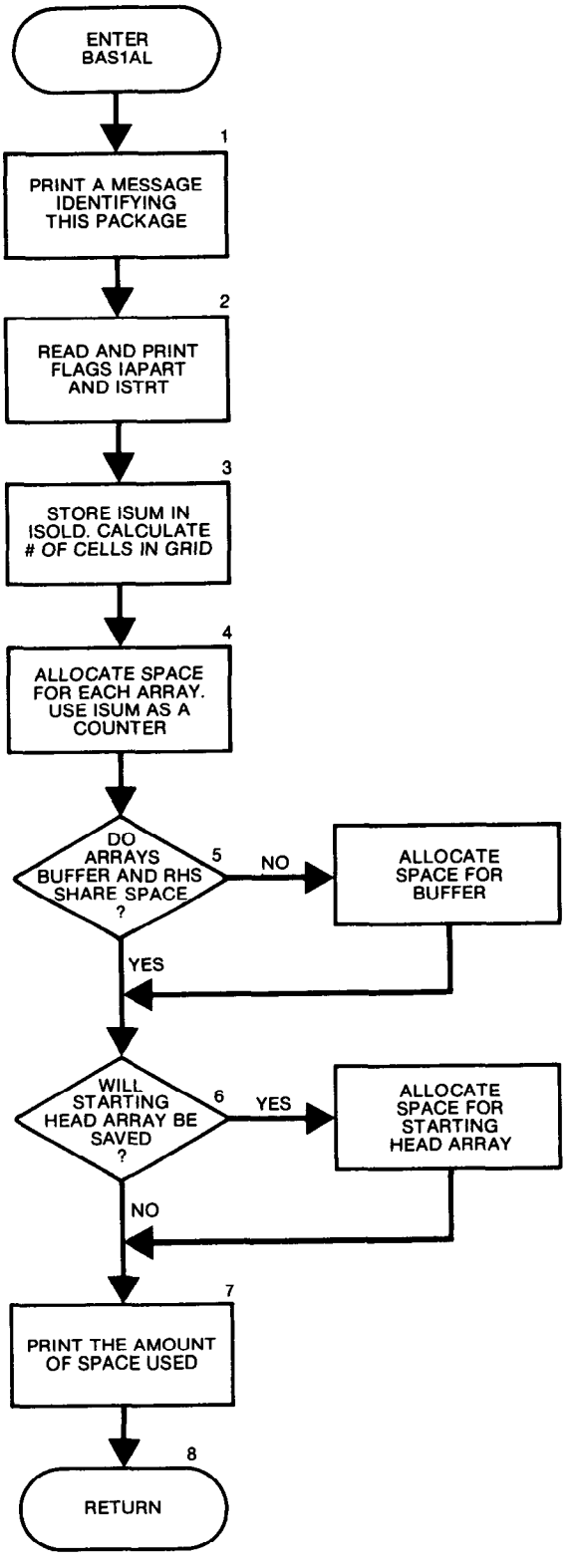
BUFFER is an array in which data is temporarily stored while it is being gathered for printing.

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

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

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

ISUM is a counter which contains the location of the first unallocated element in the X array. Each time space is allocated for an array; the value in ISUM is incremented by the size of the array.



```
      SUBROUTINE BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1            LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,INBAS,
     1            ISTRT,NCOL,NROW,NLAY,IOUT)
C-----VERSION 1515 12MAY1987 BAS1AL
C     ******************************************************************
C     ALLOCATE SPACE FOR BASIC MODEL ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------PRINT A MESSAGE IDENTIFYING THE PACKAGE.
      WRITE(IOUT,1)INBAS
    1 FORMAT(1HO,'BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 9/1/87',
     2' INPUT READ FROM UNIT',I3)
C
C2------READ & PRINT FLAG IAPART (RHS & BUFFER SHARE SPACE?) AND
C2------FLAG ISTRT (SHOULD STARTING HEADS BE SAVED FOR DRAWDOWN?)
      READ(INBAS,2) IAPART,ISTRT
    2 FORMAT(2I10)
      IF(IAPART.EQ.0) WRITE(IOUT,3)
    3 FORMAT(1X,'ARRAYS RHS AND BUFF WILL SHARE MEMORY.')
      IF(ISTRT.NE.0) WRITE(IOUT,4)
    4 FORMAT(1X,'START HEAD WILL BE SAVED')
      IF(ISTRT.EQ.0) WRITE(IOUT,5)
    5 FORMAT(1X,'START HEAD WILL NOT BE SAVED',
     1      ' -- DRAWDOWN CANNOT BE CALCULATED')
C
C3------STORE,IN ISOLD, LOCATION OF FIRST UNALLOCATED SPACE IN X.
      ISOLD=ISUM
      NRCL=NROW*NCOL*NLAY
C
C4------ALLOCATE SPACE FOR ARRAYS.
      LCHNEW=ISUM
      ISUM=ISUM+2*NRCL
      LCHOLD=ISUM
      ISUM=ISUM+NRCL
      LCIBOU=ISUM
      ISUM=ISUM+NRCL
      LCCR=ISUM
      ISUM=ISUM+NRCL
      LCCC=ISUM
      ISUM=ISUM+NRCL
      LCCV=ISUM
      ISUM=ISUM+NROW*NCOL*(NLAY-1)
      LCHCOF=ISUM
      ISUM=ISUM+NRCL
      LCRHS=ISUM
      ISUM=ISUM+NRCL
      LCDELR=ISUM
      ISUM=ISUM+NCOL
      LCDELC=ISUM
      ISUM=ISUM+NROW
      LCIOFL=ISUM
      ISUM=ISUM+NLAY*4
C
C5------IF BUFFER AND RHS SHARE SPACE THEN LCBUFF=LCRHS.
      LCBUFF=LCRHS
      IF(IAPART.EQ.0) GO TO 50
      LCBUFF=ISUM
      ISUM=ISUM+NRCL
C
C6------IF STRT WILL BE SAVED THEN ALLOCATE SPACE.
   50 LCSTRT=ISUM
      IF(ISTRT.NE.0) ISUM=ISUM+NRCL
      ISP=ISUM-ISOLD
C
C7------PRINT AMOUNT OF SPACE USED.
      WRITE(IOUT,6) ISP
    6 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED BY BAS')
      ISUM1=ISUM-1
      WRITE(IOUT,7) ISUM1,LENX
    7 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,8)
    8 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C
C8------RETURN
      RETURN
C
      END
```

List of Variables for Module BAS1AL

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

Narrative for Module BAS1RP

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

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

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

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

Module BAS1RP performs its functions in the following order:

1. Print the simulation title and calculate the number of cells in a layer.

2. Read the boundary array (IBOUND).

3. Read and print the head value to be printed for no-flow cells (HNOFLO).

4. Read the starting heads into array HOLD.

5. Copy the starting heads (and convert to double precision) from HOLD into HNEW.

6. If the starting heads must be saved, copy them from HOLD to STRT.

7. Initialize volumetric-budget accumulators.

8. Call submodule SBAS1I to initialize the Output Control System.

9. RETURN.

Flow Chart for Module BAS1RP

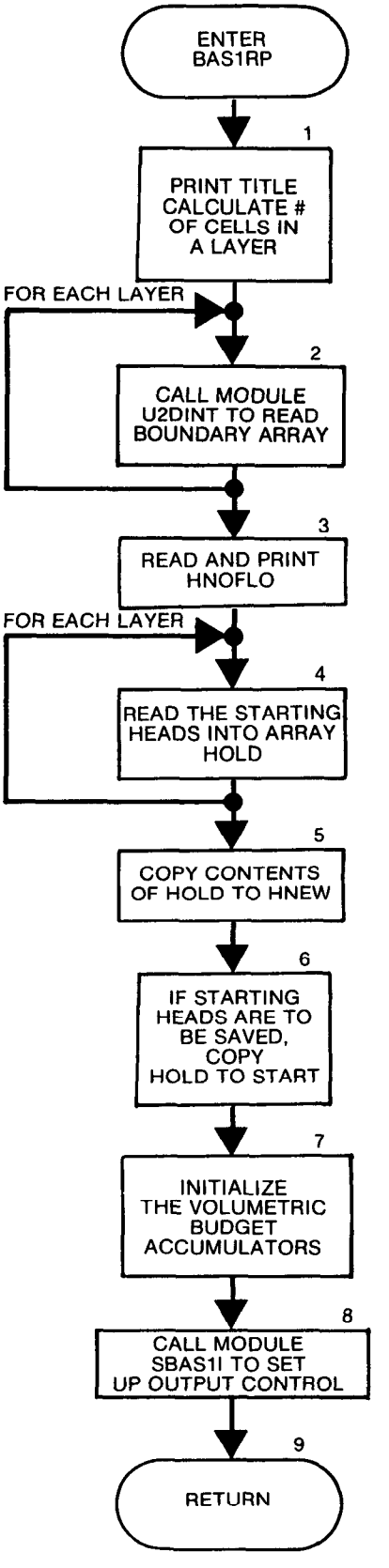
HNOFLO is a value assigned to head in inactive (no-flow) cells. It makes those cells stand out in listings of heads.

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

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

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

U2DINT is a utility module which reads two-dimensional integer arrays.



```
      SUBROUTINE BAS1RP(IBOUND,HNEW,STRT,HOLD,ISTRT,INBAS,
     1    HEADNG,NCOL,NROW,NLAY,NODES,VBVL,IOFLG,INOC,IHEDFM,
     2    IDDNFM,IHEDUN,IDDNUN,IOUT)
C-----VERSION 1628 15MAY1987 BAS1RP
C     ******************************************************************
C     READ AND INITIALIZE BASIC MODEL ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 HEADNG,ANAME
      DOUBLE PRECISION HNEW,HNOFLO
C
      DIMENSION HNEW(NODES),IBOUND(NODES),STRT(NODES),HOLD(NODES),
     1             ANAME(6,2),VBVL(4,20),IOFLG(NLAY,4),HEADNG(32)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1   ANAME(6,1) /'    ','    ','  BO','UNDA','RY A','RRAY'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1   ANAME(6,2) /'    ','    ','    ','INIT','IAL ','HEAD'/
C     ------------------------------------------------------------------
C
C1------PRINT SIMULATION TITLE, CALCULATE # OF CELLS IN A LAYER.
      WRITE(IOUT,1) HEADNG
    1 FORMAT(1H1,32A4)
      NCR=NCOL*NROW
C
C2------READ BOUNDARY ARRAY(IBOUND) ONE LAYER AT A TIME.
      DO 100 K=1,NLAY
      KK=K
      LOC=1+(K-1)*NCR
      CALL U2DINT(IBOUND(LOC),ANAME(1,1),NROW,NCOL,KK,INBAS,IOUT)
  100 CONTINUE
C
C3------READ AND PRINT HEAD VALUE TO BE PRINTED FOR NO-FLOW CELLS.
      READ(INBAS,2) TMP
    2 FORMAT(F10.0)
      HNOFLO=TMP
      WRITE(IOUT,3) TMP
    3 FORMAT(1HO,'AQUIFER HEAD WILL BE SET TO ',1PG11.5,
     1       ' AT ALL NO-FLOW NODES (IBOUND=0).')
C
C4------READ STARTING HEADS.
      DO 300 K=1,NLAY
      KK=K
      LOC=1+(K-1)*NCR
      CALL U2DREL(HOLD(LOC),ANAME(1,2),NROW,NCOL,KK,INBAS,IOUT)
  300 CONTINUE
C
C5------COPY INITIAL HEADS FROM HOLD TO HNEW.
      DO 400 I=1,NODES
      HNEW(I)=HOLD(I)
      IF(IBOUND(I).EQ.0) HNEW(I)=HNOFLO
  400 CONTINUE
C
C6------IF STARTING HEADS ARE TO BE SAVED THEN COPY HOLD TO STRT.
      IF(ISTRT.EQ.0) GO TO 590
      DO 500 I=1,NODES
      STRT(I)=HOLD(I)
  500 CONTINUE
C
C7------INITIALIZE VOLUMETRIC BUDGET ACCUMULATORS TO ZERO.
  590 DO 600 I=1,20
      DO 600 J=1,4
      VBVL(J,I)=0.
  600 CONTINUE
C
C8------SET UP OUTPUT CONTROL.
      CALL SBAS1I(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM,
     1         IDDNFM,IHEDUN,IDDNUN)
C
C9------RETURN
 1000 RETURN
      END
```

List of Variables for Module BAS1RP

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

Narrative for Module BAS1ST

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

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

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

Module BAS1ST performs its functions in the following order:

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

2. Calculate the length of the first time step.

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

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

3. Print the timing information.

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

5. RETURN.

Flow Chart for Module BAS1ST

PERLEN is the length of a stress period.

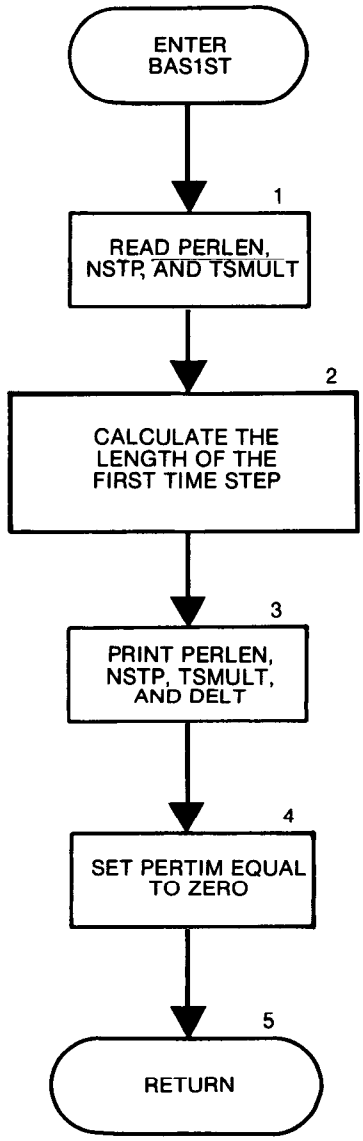
NSTP is the number of time steps in a stress period.

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

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

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

PERTIM is a field in which elapsed time during a stress period is accumulated. During each time step, the length of the time step is added to PERTIM.



```
      SUBROUTINE BAS1ST(NSTP,DELT,TSMULT,PERTIM,KPER,INBAS,IOUT)
C
C
C-----VERSION 1614 08SEP1982 BAS1ST
C     ******************************************************************
C     SETUP TIME PARAMETERS FOR NEW TIME PERIOD
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------READ LENGTH OF STRESS PERIOD, NUMBER OF TIME STEPS AND.
C1------TIME STEP MULTIPLIER.
      READ (INBAS,1) PERLEN,NSTP,TSMULT
    1 FORMAT(F10.0,I10,F10.0)
C
C2------CALCULATE THE LENGTH OF THE FIRST TIME STEP.
C
C2A-----ASSUME TIME STEP MULTIPLIER IS EQUAL TO ONE.
      DELT=PERLEN/FLOAT(NSTP)
C
C2B-----IF TIME STEP MULTIPLIER IS NOT ONE THEN CALCULATE FIRST
C2B-----TERM OF GEOMETRIC PROGRESSION.
      IF(TSMULT.NE.1.) DELT=PERLEN*(1.-TSMULT)/(1.-TSMULT**NSTP)
C
C3------PRINT TIMING INFORMATION.
      WRITE (IOUT,2) KPER,PERLEN,NSTP,TSMULT,DELT
    2 FORMAT(1H1,51X,'STRESS PERIOD NO.',I4,', LENGTH =',G15.7/52X
     1,46('-')//52X,'NUMBER OF TIME STEPS =',I6
     2//53X,'MULTIPLIER FOR DELT =',F10.3
     3//50X,'INITIAL TIME STEP SIZE =',G15.7)
C
C4------INITIALIZE PERTIM (ELAPSED TIME WITHIN STRESS PERIOD).
      PERTIM=0.
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module BAS1ST

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

Narrative for Module BAS1AD

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

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

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

Module BAS1AD performs its functions in the following order:

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

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

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

4. RETURN.

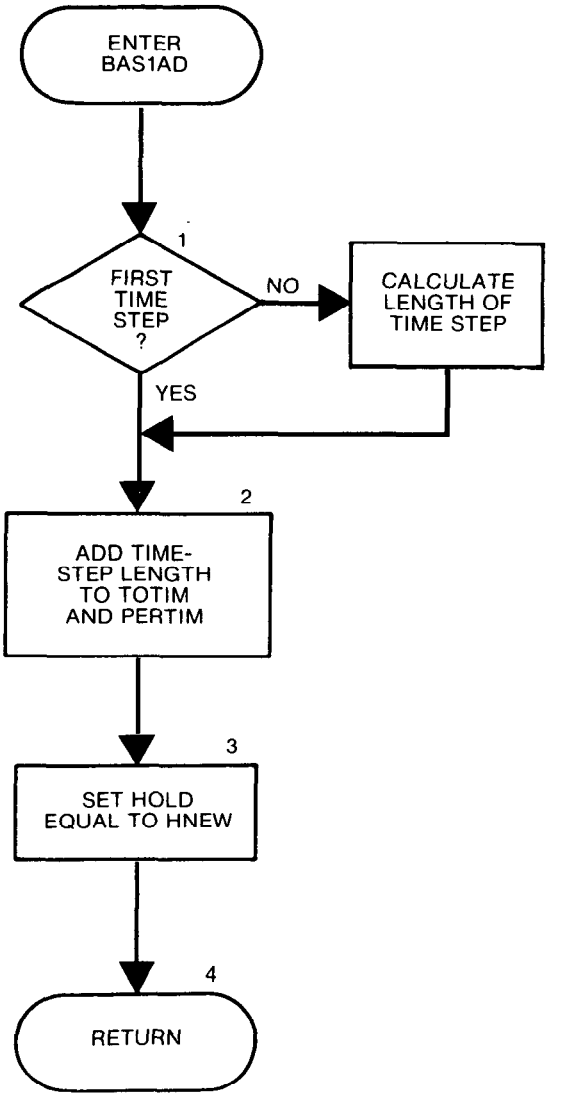
Flow Chart for Module BAS1AD

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

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

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

HNEW is the head distribution at the end of a time step.



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

List of Variables for Module BAS1AD

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

Narrative for Module BAS1FM

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

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

Module BAS1FM performs its functions in the following order:

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

2. RETURN.

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

List of Variables for Module BAS1FM

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

Narrative for Module BAS1OC

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

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

Module BAS1OC performs its functions in the following order:

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

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

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

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

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

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

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

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

9. RETURN.

Flow Chart for Module BAS1OC

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

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

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

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

If INCODE > 0, IOFLG is read.

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

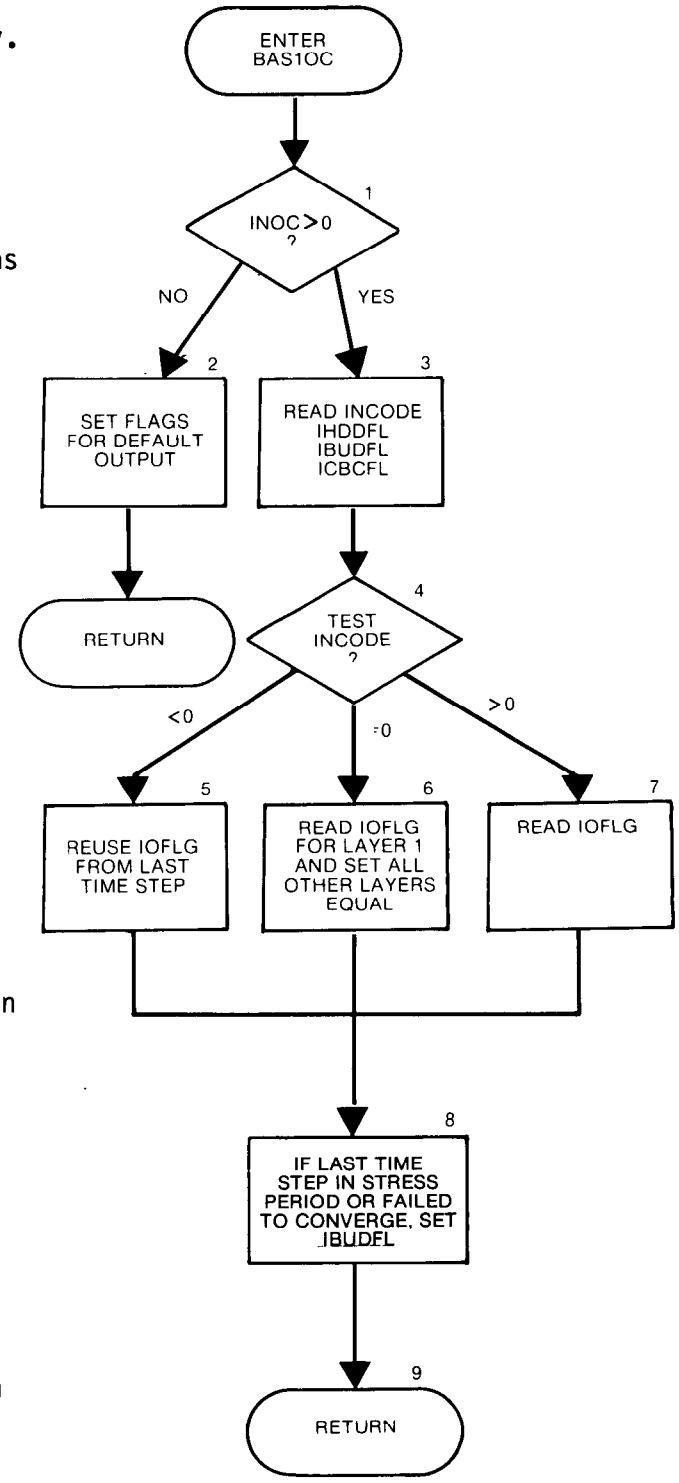
1---head print
2---drawdown print
3---head save
4---drawdown save

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

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

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

ICBCFL is the cell-by-cell flow term flag. If it is set, cell-by-cell flow terms will be printed or recorded on disk for those components of flow for which the CBC flag (IWELCB, IRCHCB, IDRNCB, etc.) is set.



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

List of Variables for Module BAS1OC

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

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.

Flow Chart for Module BAS1OT

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

SBAS1H is a submodule which writes heads.

SBAS1D is a submodule which writes drawdown.

SBAS1V is a submodule which prints the volumetric budget.

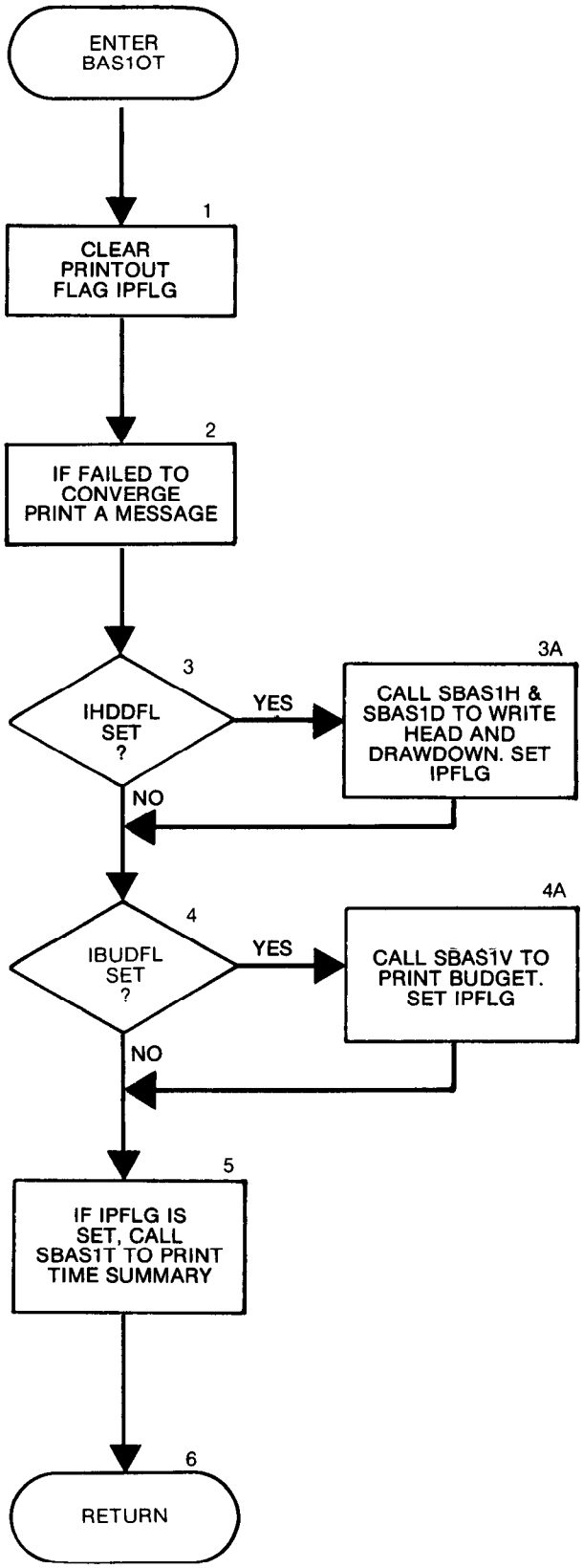
SBAS1T is a submodule which prints a time summary.

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

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

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

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



```
      SUBROUTINE 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 1522 12MAY1987 BAS1OT
C     ******************************************************************
C     OUTPUT TIME, VOLUMETRIC BUDGET, HEAD, AND DRAWDOWN
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
     1          VBNM(4,20),VBVL(4,20),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(1HO,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,4) ≠ 0, drawdown will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPFLG	Package	Flag. = 0 means nothing has been printed this time step. ≠ 0 means something has been printed this time step; therefore, a time summary must be printed.
ISTRT	Package	Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
ITMUNI	Package	Code for time units for this problem: 0 - undefined 1 - seconds 2 - minutes 3 - hours 4 - days 5 - years

List of Variables for Module 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.

Flow Chart for Module SBAS1D

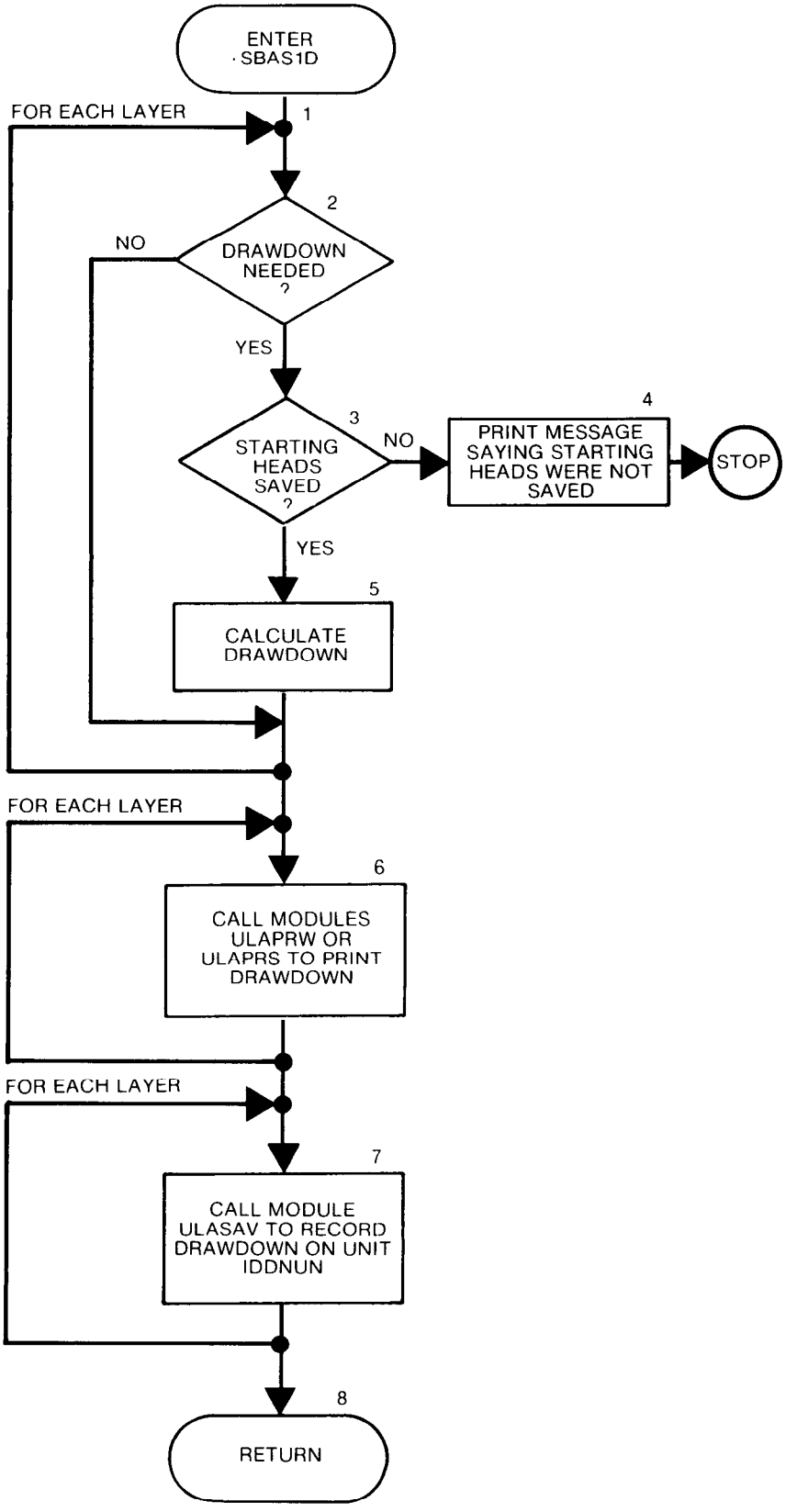
IOFLG is a table containing one entry for each layer. Each element consists of four flags which, when set, cause (1) head to be printed, (2) drawdown to be printed, (3) head to be recorded, and (4) drawdown to be recorded.

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

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

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

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



```
      SUBROUTINE SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
     1    NLAY,IOUT,IDDNFM,IDDNUN,STRT,ISTRT,IBOUND,IPFLG,
     2    PERTIM,TOTIM)
C-----VERSION 1630 15MAY1987 SBAS1D
C     ******************************************************
C     CALCULATE PRINT AND RECORD DRAWDOWNS
C     ******************************************************
C
C        SPECIFICATIONS
C     ------------------------------------------------------
      CHARACTER*4 TEXT
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),TEXT(4),
     1     BUFF(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
     2     IBOUND(NCOL,NROW,NLAY)
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','DRAW',
     1       'DOWN'/
C     ------------------------------------------------------
C
C1------FOR EACH LAYER CALCULATE DRAWDOWN IF PRINT OR RECORD
C1------IS REQUESTED
      DO 59 K=1,NLAY
C
C2------IS DRAWDOWN NEEDED FRO THIS LAYER?
      IF(IOFLG(K,2).EQ.0 .AND. IOFLG(K,4).EQ.0) GO TO 59
C
C3------DRAWDOWN IS NEEDED. WERE STARTING HEADS SAVED?
      IF(ISTRT.NE.0) GO TO 53
C
C4------STARTING HEADS WERE NOT SAVED. PRINT MESSAGE AND STOP.
      WRITE(IOUT,52)
   52 FORMAT(1HO,'CANNOT CALCULATE DRAWDOWN BECAUSE START',
     1   ' HEADS WERE NOT SAVED')
      STOP
C
C5------CALCULATE DRAWDOWN FOR THE LAYER.
   53 DO 58 I=1,NROW
      DO 58 J=1,NCOL
      HSING=HNEW(J,I,K)
      BUFF(J,I,K)=HSING
      IF(IBOUND(J,I,K).NE.0) BUFF(J,I,K)=STRT(J,I,K)-HSING
   58 CONTINUE
   59 CONTINUE
C
C6------FOR EACH LAYER: DETERMINE IF DRAWDOWN SHOULD BE PRINTED.
C6------IF SO THEN CALL ULAPRS OR ULAPRW TO PRINT DRAWDOWN.
      DO 69 K=1,NLAY
      KK=K
      IF(IOFLG(K,2).EQ.0) GO TO 69
      IF(IDDNFM.LT.0) CALL ULAPRS(BUFF(1,1,K),TEXT(1),KSTP,KPER,
     1             NCOL,NROW,KK,-IDDNFM,IOUT)
      IF(IDDNFM.GE.0) CALL ULAPRW(BUFF(1,1,K),TEXT(1),KSTP,KPER,
     1             NCOL,NROW,KK,IDDNFM,IOUT)
      IPFLG=1
   69 CONTINUE
C
C7------FOR EACH LAYER: DETERMINE IF DRAWDOWN SHOULD BE RECORDED.
C7------IF SO THEN CALL ULASAV TO RECORD DRAWDOWN.
      IFIRST=1
      IF(IDDNUN.LE.0) GO TO 80
      DO 79 K=1,NLAY
      KK=K
      IF(IOFLG(K,4).LE.0) GO TO 79
      IF(IFIRST.EQ.1) WRITE(IOUT,74) IDDNUN,KSTP,KPER
   74 FORMAT(1HO,'DRAWDOWN WILL BE SAVED ON UNIT',I3,
     1    ' AT END OF TIME STEP',I3,', STRESS PERIOD',I3)
      IFIRST=0
      CALL ULASAV(BUFF(1,1,K),TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1             NROW,KK,IDDNUN)
   79 CONTINUE
C
C8------RETURN
   80 RETURN
      END
```

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,4) ≠ 0, drawdown will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPFLG	Package	Flag. = 0 means nothing has been printed this time step. ≠ 0 means something has been printed this time step; therefore, a time summary must be printed.
ISTRT	Package	Flag. ≠ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
J	Module	Index for columns.
K	Module	Index for layers.
KK	Module	Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument, which causes problems with some compilers.
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.

Flow Chart for Module SBAS1H

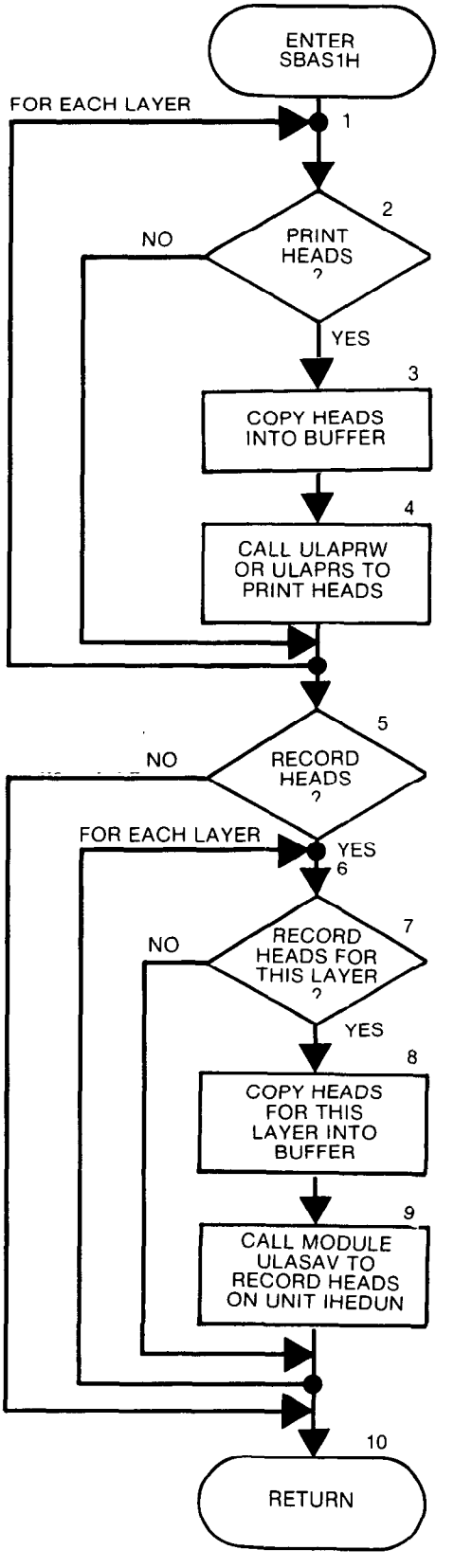
IOFLG is a table containing one entry for each layer. Each element consists of four flags which, when set, cause (1) head to be printed, (2) drawdown to be printed, (3) head to be recorded, and (4) drawdown to be recorded.

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

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

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

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



```
      SUBROUTINE SBAS1H(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,
     1    NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
C
C-----VERSION 1653 15MAY1987 SBAS1H
C     ******************************************************
C     PRINT AND RECORD HEADS
C     ******************************************************
C
C        SPECIFICATIONS
C     ------------------------------------------------------
      CHARACTER*4 TEXT
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IOFLG(NLAY,4),TEXT(4),
     1     BUFF(NCOL,NROW,NLAY)
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','    ',
     1       'HEAD'/
C     ------------------------------------------------------
C
C1------FOR EACH LAYER: PRINT HEAD IF REQUESTED.
      DO 39 K=1,NLAY
      KK=K
C
C2------TEST IOFLG TO SEE IF HEAD SHOULD BE PRINTED.
      IF(IOFLG(K,1).EQ.0) GO TO 39
      IPFLG=1
C
C3------COPY HEADS FOR THIS LAYER INTO BUFFER.
      DO 32 I=1,NROW
      DO 32 J=1,NCOL
      BUFF(J,I,1)=HNEW(J,I,K)
   32 CONTINUE
C
C4------CALL UTILITY MODULE TO PRINT CONTENTS OF BUFFER.
      IF(IHEDFM.LT.0) CALL ULAPRS(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,KK,
     1          -IHEDFM,IOUT)
      IF(IHEDFM.GE.0) CALL ULAPRW(BUFF,TEXT(1),KSTP,KPER,NCOL,NROW,KK,
     1            IHEDFM,IOUT)
   39 CONTINUE
C
C5------IF UNIT FOR RECORDING HEADS <= 0: THEN RETURN.
      IF(IHEDUN.LE.0)GO TO 50
      IFIRST=1
C
C6------FOR EACH LAYER: RECORD HEAD IF REQUESTED.
      DO 49 K=1,NLAY
      KK=K
C
C7------CHECK IOFLG TO SEE IF HEAD FOR THIS LAYER SHOULD BE RECORDED.
      IF(IOFLG(K,3).LE.0) GO TO 49
      IF(IFIRST.EQ.1) WRITE(IOUT,41) IHEDUN,KSTP,KPER
   41 FORMAT(1HO,'HEAD WILL BE SAVED ON UNIT',I3,' AT END OF TIME STEP',
     1    I3,', STRESS PERIOD',I3)
      IFIRST=0
C
C8------COPY HEADS FOR THIS LAYER INTO BUFFER.
      DO 44 I=1,NROW
      DO 44 J=1,NCOL
      BUFF(J,I,1)=HNEW(J,I,K)
   44 CONTINUE
C
C9------RECORD CONTENTS OF BUFFER ON UNIT=IHEDUN
      CALL ULASAV(BUFF,TEXT(1),KSTP,KPER,PERTIM,TOTIM,NCOL,NROW,KK,
     1            IHEDUN)
   49 CONTINUE
C
C10-----RETURN
   50 RETURN
      END
```

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) $\neq$ 0, heads will be printed. (NLAY,2) $\neq$ 0, drawdown will be printed. (NLAY,3) $\neq$ 0, heads will be recorded. (NLAY,4) $\neq$ 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. $\neq$ 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.
KK	Module	Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument, which causes problems with some compilers.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
PERTIM	Package	Elapsed time during the current stress period.
TEXT	Module	Label to be printed or recorded with array data.
TOTIM	Package	Elapsed time in the simulation.

Narrative for Module SBAS1I

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

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

Module SBAS1I performs its functions in the following order:

1. Test the unit number for Output Control (IUNIT (12)), which is known in this module by the name INOC, to see if it is positive. If it is positive, the Output Control option is active and output specification will be read from the unit number contained in INOC. If it is not positive, the Output Control option is not active and flags are set to defaults. GO TO 2 OR 3.

2. Output Control is active. Read and print the head-print format code (IHEDFM), the drawdown-print format code (IDDNFM), the unit number to record heads (IHEDUN), and the unit number to record drawdown (IDDNUN). GO TO 6. Note: The formats and associated codes are listed in the Input Instructions for Output Control.

3. Output Control is inactive. Print a message listing the defaults.

4. Set the print-format codes (IHEDFM and IDDNFM) equal to zero to get the default format.

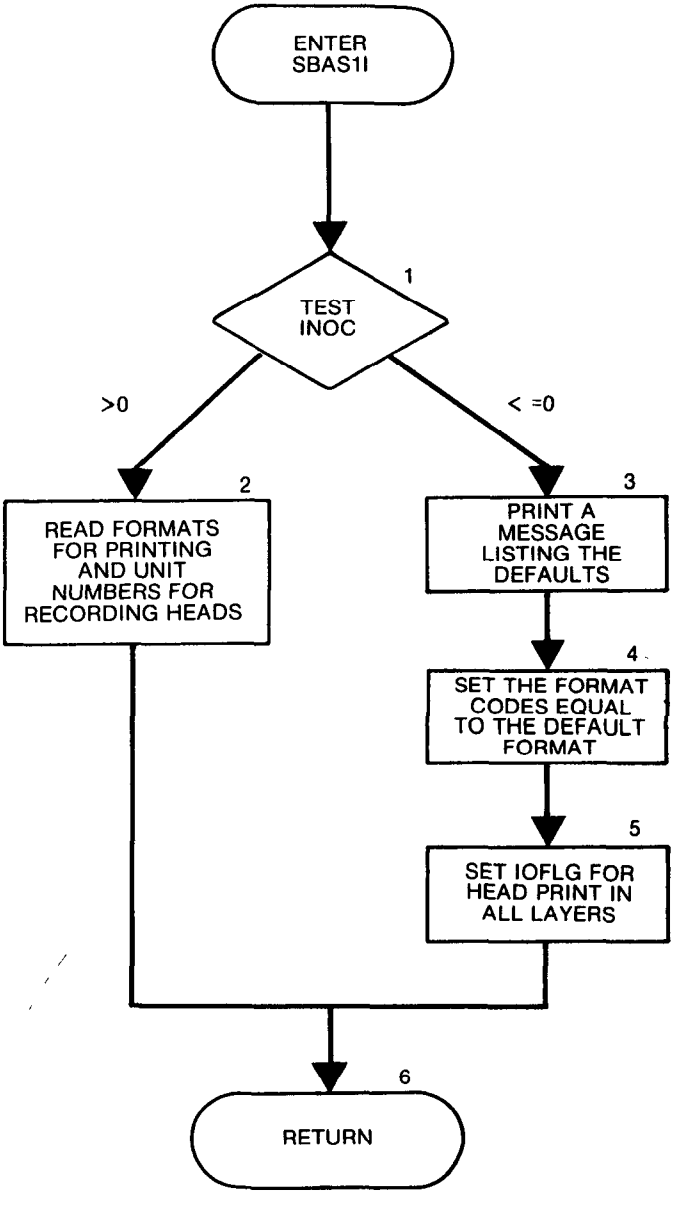
5. Set the flags in IOFLG so that head and drawdown are printed for all layers.

6. RETURN.

Flow Chart for Module SBAS1I

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

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



```
      SUBROUTINE SBAS1I(NLAY,ISTRT,IOFLG,INOC,IOUT,IHEDFM,
     1          IDDNFM,IHEDUN,IDDNUN)
C
C-----VERSION 1531 12MAY1987 SBAS1I
C     ***************************************************************
C     SET UP OUTPUT CONTROL
C     ***************************************************************
C
C        SPECIFICATIONS:
C     -------------------------------------------------------------------
      DIMENSION IOFLG(NLAY,4)
C     -------------------------------------------------------------------
C
C1------TEST UNIT NUMBER FROM IUNIT (INOC) TO SEE IF OUTPUT
C1------CONTROL IS ACTIVE.
      IF(INOC.LE.0) GO TO 600
C
C2------READ AND PRINT FORMATS FOR PRINTING AND UNIT NUMBERS FOR
C2------RECORDING HEADS AND DRAWDOWN. THEN RETURN.
  500 READ (INOC,1)IHEDFM,IDDNFM,IHEDUN,IDDNUN
    1 FORMAT (4I10)
      WRITE (IOUT,3)IHEDFM,IDDNFM
    3 FORMAT (1H0,'HEAD PRINT FORMAT IS FORMAT NUMBER',I4,
     1        '    DRAWDOWN PRINT FORMAT IS FORMAT NUMBER',I4)
      WRITE (IOUT,4)IHEDUN,IDDNUN
    4 FORMAT (1H0,'HEADS WILL BE SAVED ON UNIT',I3,
     1        '    DRAWDOWNS WILL BE SAVED ON UNIT',I3)
      WRITE(IOUT,561)
  561 FORMAT(1H0,'OUTPUT CONTROL IS SPECIFIED EVERY TIME STEP')
      GO TO 1000
C
C3------OUTPUT CONTROL IS INACTIVE. PRINT A MESSAGE LISTING DEFAULTS.
  600 WRITE(IOUT,641)
  641 FORMAT(1H0,'DEFAULT OUTPUT CONTROL -- THE FOLLOWING OUTPUT',
     1       ' COMES AT THE END OF EACH STRESS PERIOD:')
      WRITE(IOUT,642)
  642 FORMAT(1X,'TOTAL VOLUMETRIC BUDGET')
      WRITE(IOUT,643)
  643 FORMAT(1X,10X,'HEAD')
      IF(ISTRT.NE.0)WRITE(IOUT,644)
  644 FORMAT(1X,10X,'DRAWDOWN')
C
C4------SET THE FORMAT CODES EQUAL TO THE DEFAULT FORMAT.
      IHEDFM=0
      IDDNFM=0
C
C5------SET DEFAULT FLAGS IN IOFLG SO THAT HEAD (AND DRAWDOWN) IS
C5------PRINTED FOR EVERY LAYER.
      ID=0
      IF(ISTRT.NE.0) ID=1
  670 DO 680 K=1,NLAY
      IOFLG(K,1)=1
      IOFLG(K,2)=ID
      IOFLG(K,3)=0
      IOFLG(K,4)=0
  680 CONTINUE
      GO TO 1000
C
C6------RETURN
1000  RETURN
      END
```

List of Variables for Module SBAS1I

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.
IOFLG	Package	DIMENSION (NLAY,4), Flags to control printing and recording of head and drawdown for each layer. (NLAY,1) $\neq$ 0, heads will be printed. (NLAY,2) $\neq$ 0, drawdown will be printed. (NLAY,3) $\neq$ 0, heads will be recorded. (NLAY,4) $\neq$ 0, drawdown will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISTRT	Package	Flag. $\neq$ 0, starting heads will be saved so that drawdown can be calculated. = 0, starting heads will not be saved.
K	Module	Index for layers
NLAY	Global	Number of layers in the grid.

Narrative for Module SBAS1T

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

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

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

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

   (b) RETURN.

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

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

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

6. RETURN.

```
      SUBROUTINE SBAS1T(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)
C
C
C-----VERSION 0837 09APR1982 SBAS1T
C     ***************************************************************
C     PRINT SIMULATION TIME
C     ***************************************************************
C
C        SPECIFICATIONS:
C     ---------------------------------------------------------------
C     ---------------------------------------------------------------
      WRITE(IOUT,199) KSTP,KPER
  199 FORMAT(1H0,///10X,'TIME SUMMARY AT END OF TIME STEP',I3,
     1     ' IN STRESS PERIOD',I3)
C
C1------USE TIME UNIT INDICATOR TO GET FACTOR TO CONVERT TO SECONDS.
      CNV=0.
      IF(ITMUNI.EQ.1) CNV=1.
      IF(ITMUNI.EQ.2) CNV=60.
      IF(ITMUNI.EQ.3) CNV=3600.
      IF(ITMUNI.EQ.4) CNV=86400.
      IF(ITMUNI.EQ.5) CNV=31557600.
C
C2------IF FACTOR=0 THEN TIME UNITS ARE NON-STANDARD.
      IF(CNV.NE.0.) GO TO 100
C
C2A-----PRINT TIMES IN NON-STANDARD TIME UNITS.
      WRITE(IOUT,301) DELT,PERTIM,TOTIM
  301 FORMAT(21X,'     TIME STEP LENGTH =',G15.6/
     1       21X,'   STRESS PERIOD TIME =',G15.6/
     2       21X,'TOTAL SIMULATION TIME =',G15.6)
C
C2B-----RETURN
      RETURN
C
C3------CALCULATE LENGTH OF TIME STEP & ELAPSED TIMES IN SECONDS.
  100 DELSEC=CNV*DELT
      TOTSEC=CNV*TOTIM
      PERSEC=CNV*PERTIM
C
C4------CALCULATE TIMES IN MINUTES,HOURS,DAYS AND YEARS.
      DELMN=DELSEC/60.
      DELHR=DELMN/60.
      DELDY=DELHR/24.
      DELYR=DELDY/365.25
      TOTMN=TOTSEC/60.
      TOTHR=TOTMN/60.
      TOTDY=TOTHR/24.
      TOTYR=TOTDY/365.25
      PERMN=PERSEC/60.
      PERHR=PERMN/60.
      PERDY=PERHR/24.
      PERYR=PERDY/365.25
C
C5------PRINT TIME STEP LENGTH AND ELAPSED TIMES IN ALL TIME UNITS.
      WRITE(IOUT,200)
  200 FORMAT(27X,'     SECONDS        MINUTES         HOURS',10X,
     1    'DAYS           YEARS'/27X,75('-'))
      WRITE (IOUT,201) DELSEC,DELMN,DELHR,DELDY,DELYR
  201 FORMAT(1X,'     TIME STEP LENGTH',5X,5G15.6)
      WRITE(IOUT,202) PERSEC,PERMN,PERHR,PERDY,PERYR
  202 FORMAT(1X,'   STRESS PERIOD TIME',5X,5G15.6)
      WRITE(IOUT,203) TOTSEC,TOTMN,TOTHR,TOTDY,TOTYR
  203 FORMAT(1X,'TOTAL SIMULATION TIME',5X,5G15.6)
C
C6------RETURN
      RETURN
      END
```

List of Variables for Module SBAS1T

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.

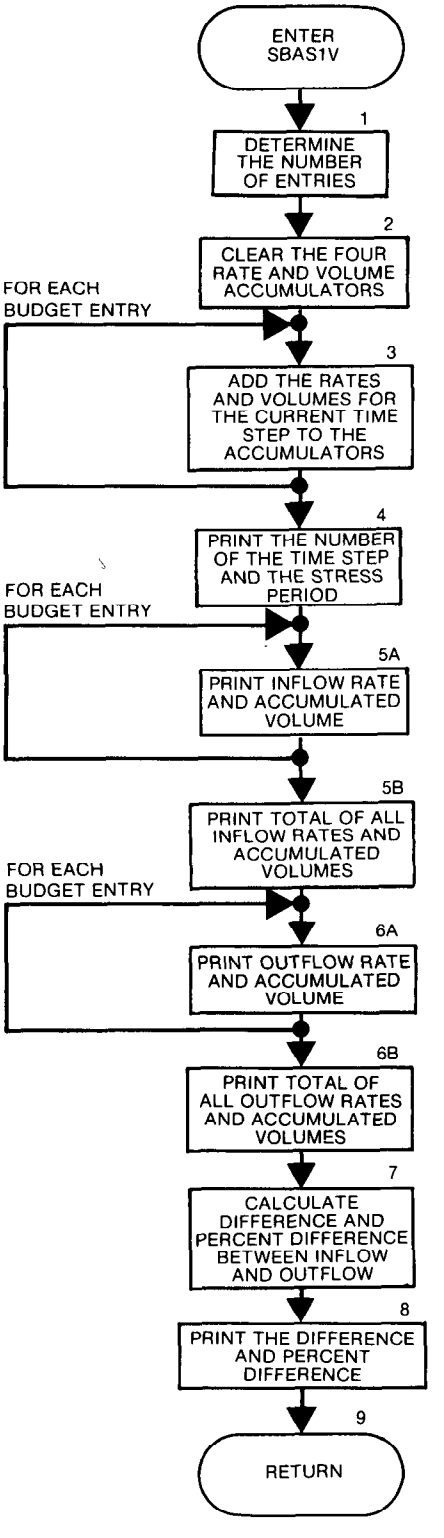
6. Print the individual output rates and volumes and their totals.

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.

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



```
      SUBROUTINE SBAS1V(MSUM,VBNM,VBVL,KSTP,KPER,IOUT)
C
C
C-----VERSION 1531 12MAY1987 SBAS1V
C     ******************************************************************
C     PRINT VOLUMETRIC BUDGET
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM
      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 determine flow between adjacent cells. It also computes the terms that determine 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; thus the name Block-Centered Flow is given to the package.

In Chapter 2, the equation of flow for each cell in the model was developed as

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

The CV, CR, and CC coefficients are conductances between nodes--sometimes called "branch conductances." The HCOF and RHS coefficients are composed of external source terms and storage terms. Besides calculating the conductances and storage terms, the BCF Package calculates flow-correction terms that are added to HCOF and RHS when an underlying aquifer becomes partially unsaturated. Under these conditions the flow to the underlying aquifer no longer increases in proportion to the head difference between aquifers, but rather reaches a constant limiting value. The additional terms correct the flow equations, in effect reducing the expressions for downward flow to correspond to this limiting value.

The following discussion of the conceptualization and implementation of the BCF package is divided into nine sections: Basic Conductance Equations, Horizontal Conductance Under Confined Conditions, Horizontal Conductance Under Water Table Conditions, Vertical Conductance Formulation, Vertical Flow Calculation Under Desaturating Conditions, Storage Formulation, Storage Term Conversion, Applicability and Limitations of Optional Formulations and Data Requirements.

Basic Conductance Equations

The concept of hydraulic conductance was introduced in Chapter 2 (equation (9)). It is reviewed here and extended to cover the calculation of equivalent conductance for elements arranged in series.

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 (figure 23) as

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

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);

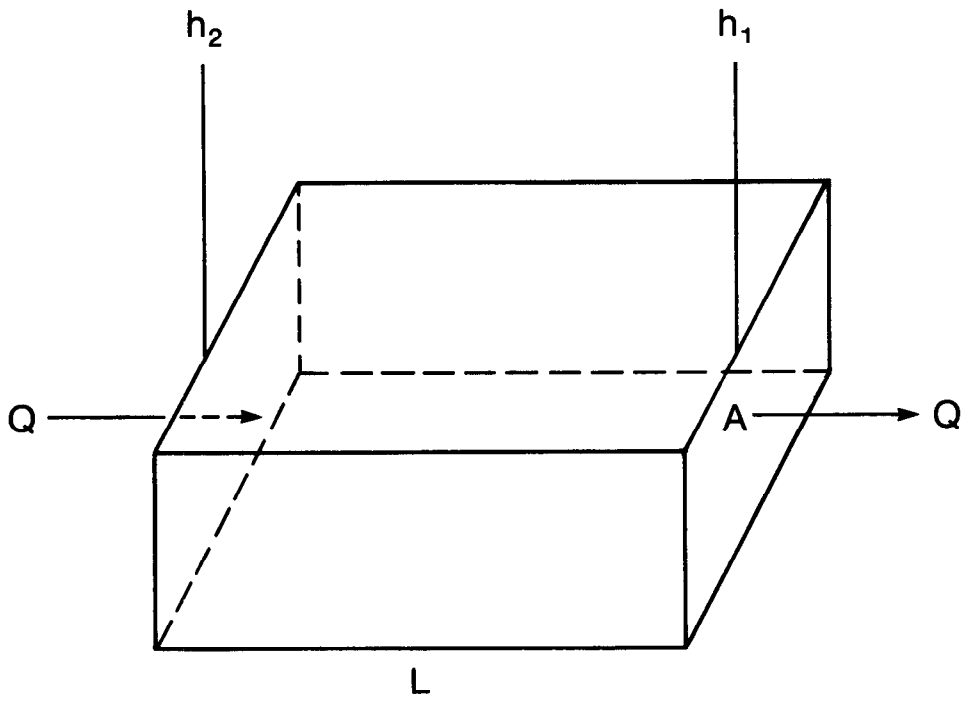
h_2-h_1 is the head differences across the prism parallel to flow (L); and

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

Conductance, C, is defined as

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

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



Explanation

K	Is Hydraulic Conductivity
h_2	Is the Head at the Left End of the Prism
h_1	Is the Head at the Right End of the Prism
Q	Is the Flow Rate from the Left End to the Right End
L	Is the Length of the Flow Path
A	Is the Cross Sectional Area Perpendicular to the Direction of Flow

Figure 23.—Prism of porous material illustrating Darcy's law.

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. In an anisotropic medium characterized by three principal directions of hydraulic conductivity, the conductances of a prism in these three principal directions will generally differ.

If a prism of porous material consists of two or more subprisms in series--that is, aligned sequentially in the direction of flow, as shown in figure 24--and the conductance of each subprism is known, a conductance representing the entire prism can be calculated. The equivalent conductance for the entire prism is the rate of flow in the prism divided by the head change across the prism.

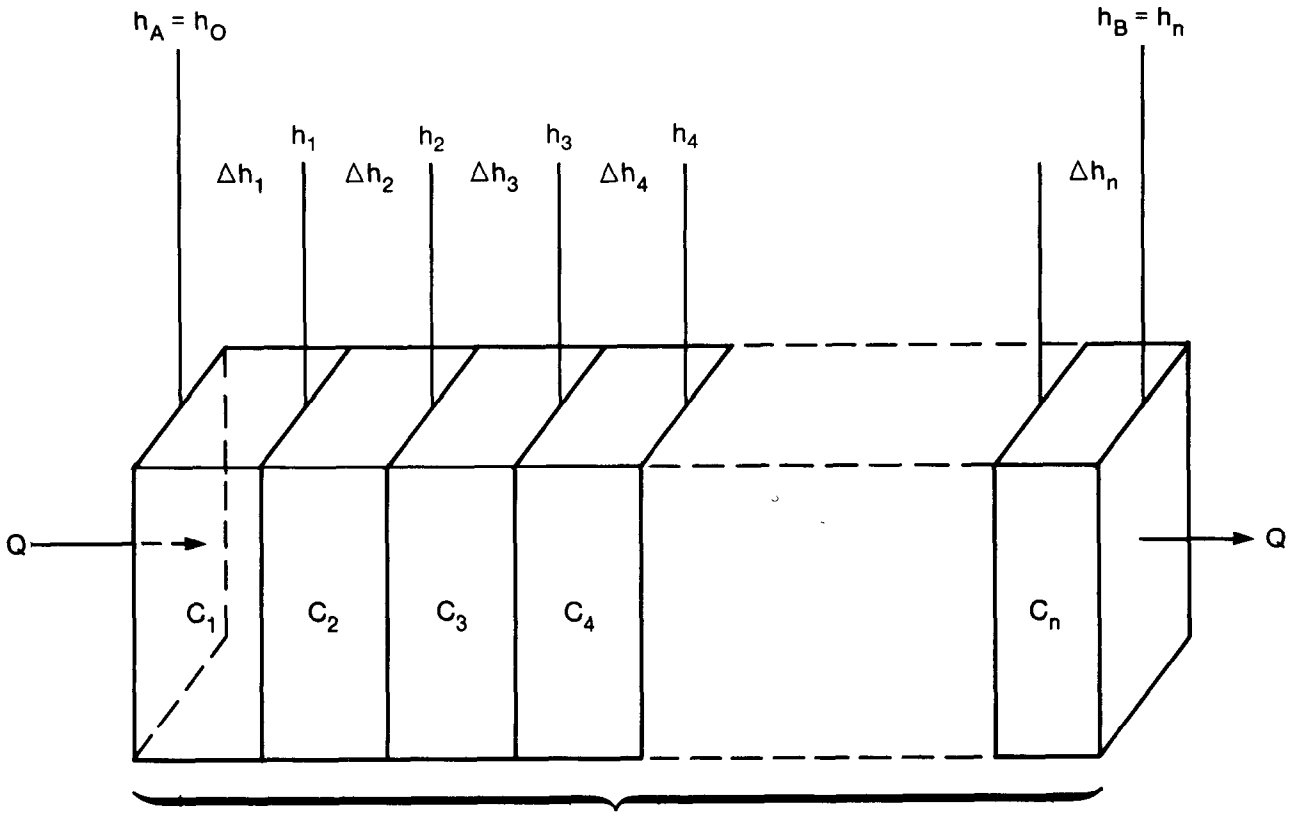
$$C = Q/(h_A-h_B) \tag{34}$$

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 (equation (32)) gives

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



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

Explanation

Q	Is the Flow Rate
C_m	Is Conductance of Prism m
h_m	Is Head at the Right Side of Prism m
Δh_m	Is the Head Change Across Prism m
C	Is the Conductance of the Entire Prism

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

Since flow is one-dimensional and we are assuming no accumulation or depletion in storage, all q_i are equal to the 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

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

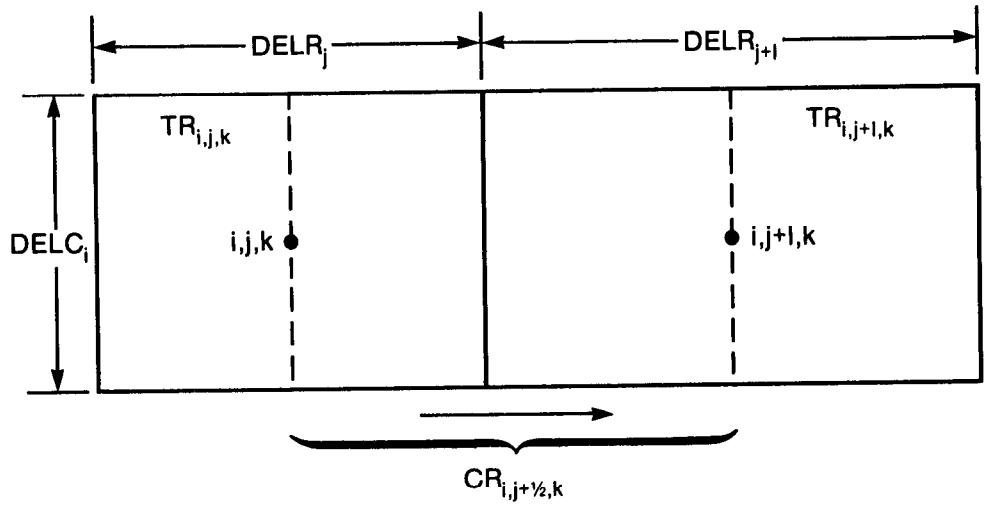
Thus for a set of conductances arranged in series, the inverse of the equivalent conductance equals the sum of the inverses of the individual conductances. When there are only two sections, the equivalent conductance reduces to

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

Horizontal Conductance Under Confined Conditions

The finite-difference equations presented in this report use equivalent conductances between nodes of adjacent cells--i.e., "branch conductances,"--rather than conductances defined within individual cells. The horizontal conductance terms, CR and CC of equation (29), are calculated between adjacent horizontal nodes. 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 designate conductance between nodes, as opposed to conductance within a cell, 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 the conductance between nodes. Two assumptions are made: (1)



$$\frac{1}{CR_{i,j+½,k}} = \frac{1}{\left(\frac{TR_{i,j,k}\,DELC_i}{\left(\frac{DELR_j}{2}\right)}\right)} + \frac{1}{\left(\frac{TR_{i,j+1,k}\,DELC_i}{\left(\frac{DELR_{j+1}}{2}\right)}\right)}$$

$$CR_{i,j+½,k} = 2\,DELC_i \times \frac{TR_{i,j,k}\,TR_{i,j+1,k}}{TR_{i,j,k}\,DELR_{j+1} + TR_{i,j+1,k}\,DELR_j}$$

Explanation

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

$CR_{i,j+½,k}$ Is Conductance in the Row Direction Between Nodes i,j,k and i,j+1,k

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

the nodes are in the center of the cells and (2) the transmissivity is uniform over each 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 the conductance for each half cell from equation (33) gives

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

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).

DELR and DELC are identical to the terms Δr and Δc, respectively, which were introduced in figure 4 and equation (3), Chapter 2. The new notation is introduced here to conform to the input of the Block-Centered Flow Package.

Simplification of the above expression gives the final equation

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

The same process can be applied to the calculation of $CC_{i+1/2,j,k}$ 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}). Equations (41) and (42) are used in the BCF Package to calculate the horizontal conductances between nodes within each layer of the model. However, where the transmissivity of both cells is zero, the conductance between the nodes in the cells is set equal to zero without invoking the equations.

Horizontal Conductance Under Water Table Conditions

In a model layer which is confined, horizontal conductance will be constant for the simulation. If a layer is unconfined or 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 as the product of hydraulic conductivity and saturated thickness; then conductance is calculated from transmissivity and cell dimensions using equations (41) and (42).

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

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

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

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

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

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

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

where

$HYR_{i,j,k}$ is the hydraulic conductivity of cell i,j,k in the row direction (Lt^{-1}); (this notation is introduced here to conform to the input of the Block-Centered Flow Package);

$TOP_{i,j,k}$ 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 (equation (45)), the cell is considered to be dewatered, and is permanently set to no flow; the model has no provision for the resaturation of a dewatered cell. Thus errors may arise in attempts to simulate situations in which actual reversals in water-level occur. Errors can also arise if oscillations of computed heads occur during iteration; if such computational oscillations cause head to drop erroneously below the botton of the cell, the cell will change to no flow for all succeeding iterations and time steps. As a means of controlling this problem, the iterative solvers contain provisions for slowing the rate of convergence.

In the program described herein a layer-type flag, LAYCON, is used to specify whether or not the simulation of water table conditions through equations (43)-(45) is to be invoked. This is discussed more fully in the section on data requirements.

Vertical Conductance Formulation

Vertical conductance terms are calculated within the model using data from an input array which incorporates both thickness and vertical hydraulic conductivity in a single term, and using horizontal (or map) areas calculated from cell dimensions. In general, the vertical interval between two nodes, i,j,k and and i,j,k+1, may be considered to contain n geohydrologic layers or units, having vertical hydraulic conductivities $K_1, K_2 \ldots K_n$ and thicknesses $\Delta z_1, \Delta z_2 \ldots \Delta z_n$. The map area of the cells around nodes i,j,k and i,j,k+1 is $DELR_j * DELC_i$; the vertical conductance of an individual geohydrologic layer, g, in this area is given by

$$C_g = \frac{K_g \, DELR_j * DELC_i}{\Delta z_g} \tag{46}$$

The equivalent vertical conductance, $C_{i,j,k+1/2}$, for the full vertical interval between nodes i,j,k and i,j,k+1 is found by treating the n individual geohydrologic layers as conductances in series; this yields

$$\frac{1}{C_{i,j,k+1/2}} = \sum_{g=1}^{n} \frac{1}{C_g} =$$

$$\sum_{g=1}^{n} \frac{1}{\frac{K_g DELR_j * DELC_i}{\Delta z_g}} = \frac{1}{DELR_j * DELC_i} * \sum_{g=1}^{n} \frac{\Delta z_g}{K_g} \tag{47}$$

rearranging equation (47)

$$\frac{C_{i,j,k+1/2}}{DELR_j * DELC_i} = \frac{1}{\sum_{g=1}^{n} \frac{\Delta z_g}{K_g}} \tag{48}$$

The quantity $\frac{C_{i,j,k+1/2}}{DELR_j * DELC_i}$ has been termed the "vertical leakance " and is designated $Vcont_{i,j,k+1/2}$ in this report; thus we have

$$Vcont_{i,j,k+1/2} = \frac{1}{\sum_{g=1}^{n} \frac{\Delta z_g}{K_g}} \qquad (49)$$

Vcont is the term actually used as input in the model described herein. That is, rather than specifying a total thickness and an equivalent (or harmonic mean) vertical hydraulic conductivity for the interval between node i,j,k and node i,j,k+1, the user specifies the term $Vcont_{i,j,k+1/2}$, which is actually the conductance of the interval divided by the cell area, and as such incorporates both hydraulic conductivity and thickness. The program multiples Vcont by cell area to obtain vertical conductance. The values of Vcont must be calculated or determined externally to the program; this is generally done through an application of equation (49). The Vcont values are actually read as the elements of a two-dimensional input array, $Vcont_{i,j}$, for each layer. Each value of $Vcont_{i,j}$ is the vertical leakance for the interval between cell i,j,k and cell i,j,k+1--that is, for the interval between the layer for which the array is read, and the layer below it. It follows that the Vcont array is not read for the lowermost layer in the model. Although values of Vcont are thus read into the model through a series of two-dimensional input arrays, the discussion in this section will continue to be given in terms of three-dimensional array notation, $Vcont_{i,j,k+1/2}$, to emphasize the fact that the Vcont values refer to the intervals between layers.

Figure 26 shows a situation in which nodes i,j,k and i,j,k+1 both fall within a single hydrogeologic unit, having a vertical hydraulic conductivity $K_{z\ i,j}$ which is uniform at least within the cell area. For this case, application of equation (49) yields

$$Vcont_{i,j,k+1/2} = \frac{K_{z\ i,j}}{\Delta z_{k+1/2}} \qquad (50)$$

where $\Delta z_{k+1/2}$, the vertical distance between nodes, is the sum of $\frac{\Delta v_k}{2}$ and $\frac{\Delta v_{k+1}}{2}$, in which Δv represents layer thickness as in figure 1. This situation might be found, for example, where several model layers are used to represent a single geohydrologic unit in order to provide greater vertical resolution.

Figure 27 shows a case in which two adjacent model layers are used to represent two vertically adjacent hydrogeologic units, so that nodes i,j,k and i,j,k+1 fall at the midpoints of these geohydrologic layers. Each layer is characterized by its own value of vertical hydraulic conductivity, which is again assumed to be uniform at least over the cell area. The expression for Vcont in this case becomes

$$Vcont_{i,j,k+1/2} = \frac{1}{\frac{(\Delta v_k)/2}{K_{z\ i,j,k}} + \frac{(\Delta v_{k+1})/2}{K_{z\ i,j,k+1}}} \qquad (51)$$

where Δv_k is the thickness of model layer k

Δv_{k+1} is the thickness of model layer k+1

$K_{z\ i,j,k}$ is the vertical hydraulic conductivity of the upper layer in cell i,j,k

$K_{z\ i,j,k+1}$ is the vertical hydraulic conductivity of the lower layer in cell i,j,k+1

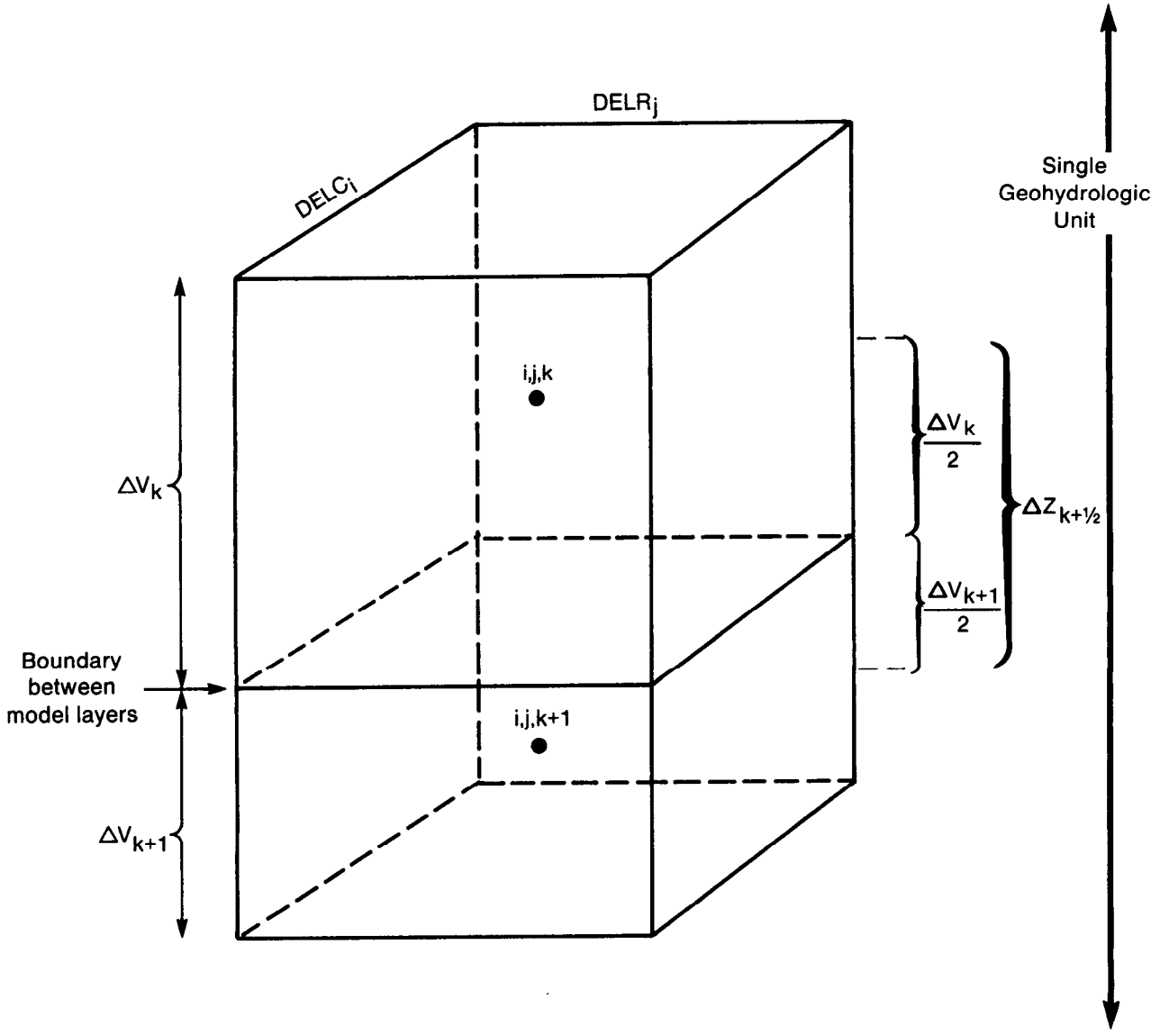


Figure 26.—Diagram for calculation of vertical leakance, Vcont, between two nodes which fall within a single geohydrologic unit.

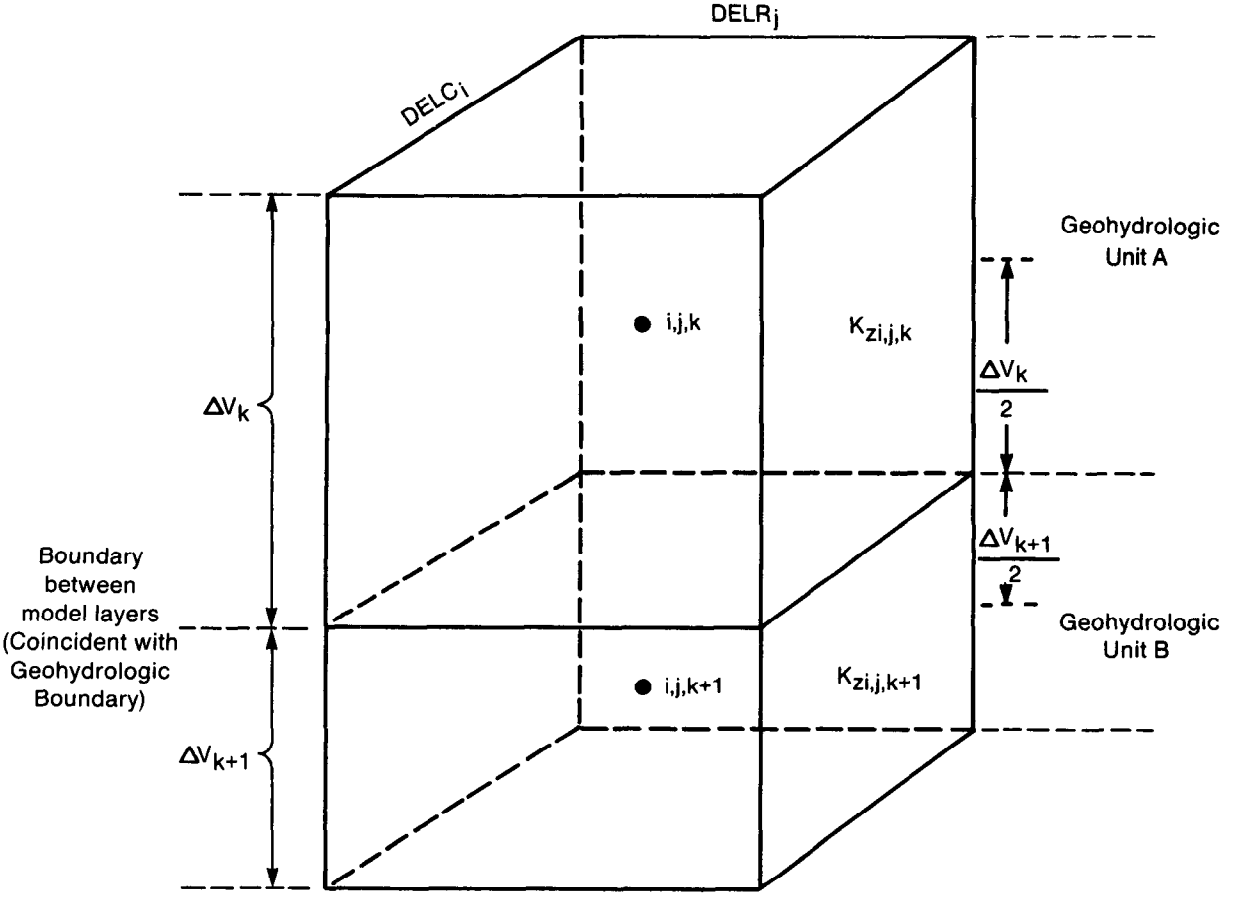


Figure 27.—Diagram for calculation of vertical leakance, Vcont, between two nodes located at the midpoints of vertically adjacent geohydrologic units.

If one value of K_z is much smaller than the other, the term containing the larger K_z value will be negligible in equation (51). Thus for this condition, only the term involving the smaller K_z value need be retained in the denominator of (51).

Figure 28 shows a third situation, in which node i,j,k and node i,j,k+1 are taken within (i.e., at the median depths of) two aquifers which are separated by a semiconfining unit. In this case, three intervals must be represented in the summation of equation (49)--the lower half of the upper aquifer, the semiconfining unit, and the upper half of the lower aquifer. The resulting expression for Vcont is

$$Vcont_{i,j,k+1/2} = \frac{1}{\frac{\Delta z_{u/2}}{K_{zu}} + \frac{\Delta z_c}{K_{zc}} + \frac{\Delta z_{L/2}}{K_{zL}}} \qquad (52)$$

where Δz_u is the thickness of the upper aquifer

Δz_c is the thickness of the confining bed

Δz_L is the thickness of the lower aquifer

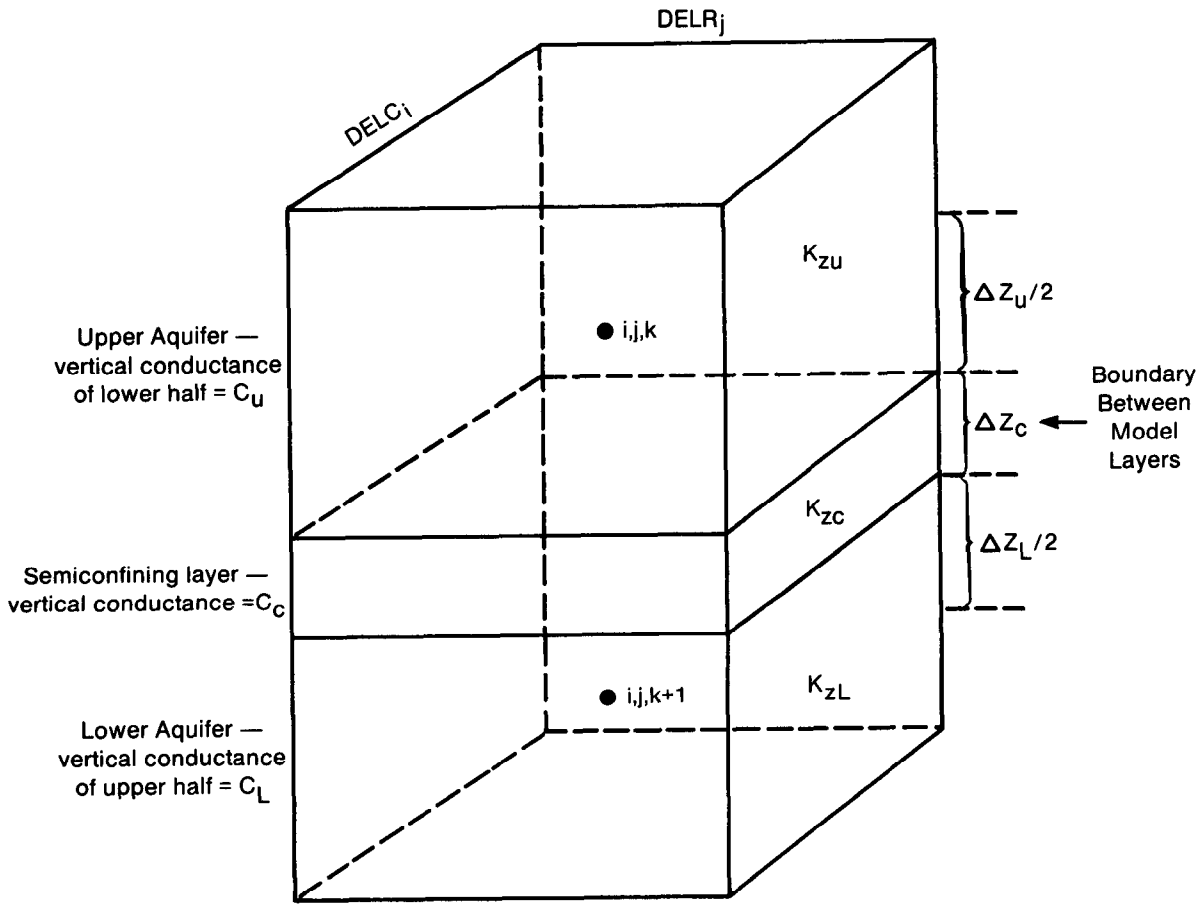
K_{zu} is the vertical hydraulic conductivity of the upper aquifer

K_{zc} is the vertical hydraulic conductivity of the semiconfining unit

K_{zL} is the vertical hydraulic conductivity of the lower aquifer; and

each of these terms must in general be considered to vary with the map location (i,j) of the nodes. In many applications it turns out that K_{zc} is much smaller than either K_{zu} or K_{zL}; in these situations the terms involving K_{zu} and K_{zL} are negligible in equation (52) so that the expression for Vcont becomes

$$Vcont_{i,j,k+1/2} = \frac{K_{zc}}{\Delta z_c} \qquad (53)$$



$$\frac{1}{C_{eq}} = \frac{1}{C_u} + \frac{1}{C_c} + \frac{1}{C_L} =$$

$$\frac{1}{DELC_i * DELR_j} \left\{ \frac{\Delta Z_u/2}{K_{zu}} + \frac{\Delta Z_c}{K_{zc}} + \frac{\Delta Z_L/2}{K_{zL}} \right\}$$

$$VCONT_{i,j,k+1/2} = \frac{1}{\frac{\Delta Z_u/2}{K_{zu}} + \frac{\Delta Z_c}{K_{zc}} + \frac{\Delta Z_L/2}{K_{zL}}}$$

Figure 28.—Diagram for calculation of vertical leakance, Vcont, between two nodes located at the midpoints of aquifers which are separated by a semiconfining unit.

If the formulation of equation (53) is applied to the situation shown in figure 28, and if the further assumptions are made that the confining bed makes no measureable contribution to the horizontal conductance or the storage capacity of either model layer, then in effect model layer k represents the upper aquifer, model layer k+1 represents the lower aquifer, and the confining bed is treated simply as the vertical conductance between the two model layers. This formulation is equivalent to that of figure 12, and is frequently referred to as the "quasi-three-dimensional" approach.

In summary, the model described herein utilizes a single input array, Vcont, which incorporates both vertical hydraulic conductivity and thickness, rather than independent inputs for thickness and conductivity. The program multiplies Vcont by cell area to obtain vertical conductance. This requires the user to calculate Vcont values externally to the program, using equation (49) in the general case (where n hydrogeologic layers occur in the vertical interval between nodes) or equations (50), (51), (52) or (53) in the situations shown in figures 26-28. While this approach involves some preprocessing of input data, it actually increases the flexibility of model application. Because layer transmissivity (or hydraulic conductivity and bottom elevation if unconfined) and layer storage coefficient are also used as input terms, the model never actually reads vertical gird spacing data. Thus the model can implement either the orthogonal mesh of figure 9-b or a deformed mesh such as that of figure 9-c, and can similarly be adapted to either a direct three-dimensional simulation or to the quasi-three-dimensional formulation, without modification of the program.

Vertical Flow Calculation Under Dewatered Conditions

The basic finite difference equation for cell i,j,k (equation (24)) was given as

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

In this equation the term $CV_{i,j,k+1/2}(h^{m}_{i,j,k+1} - h^{m}_{i,j,k})$ gives the flow into cell i,j,k through its lower face, i.e.

$$q_{i,j,k+1/2} = CV_{i,j,k+1/2}(h^{m}_{i,j,k+1} - h^{m}_{i,j,k}) \qquad (55)$$

where following the convention of equation (24), a positive value of $q_{i,j,k+1/2}$ indicates flow into cell i,j,k and a negative value indicates flow out of the cell. Equations (54) and (55) are based on the assumption that cells i,j,k and i,j,k+1 are fully saturated - i.e., that the water level in each cell stands higher than the elevation of the top of the cell. There are, however, situations in which a portion of a confined aquifer may become unsaturated--for example, when drawdown due to pumpage causes water levels to fall, at least locally, below the top of the aquifer. In terms of simulation, this condition is shown in figure 29. Two aquifers separated by a confining bed are simulated using the quasi-three-dimensional approach, in which the upper aquifer is represented by cell i,j,k, the underlying aquifer by cell i,j,k+1, and the confining bed by the vertical conductance between the two layers, $CV_{i,j,k+1/2}$. Pumping from the lower layer has

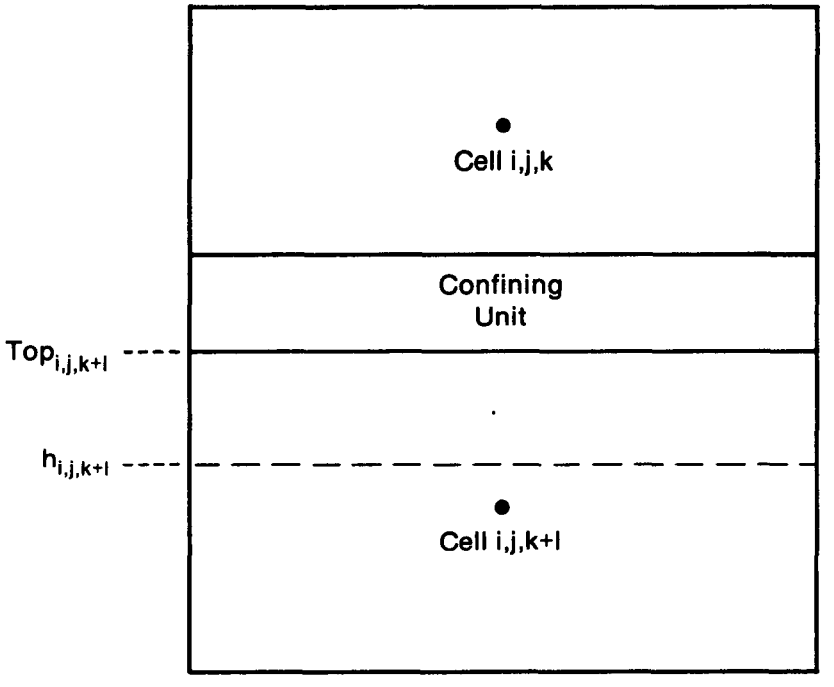


Figure 29.—Situation in which a correction is required to limit the downward flow into cell i,j,k+1, as a result of partial desaturation of the cell.

lowered the water level in cell i,j,k+1 below the elevation of the top of the cell, so that the aquifer is effectively unconfined within the cell area. An assumption is made that the confining layer remains fully saturated from top to bottom, and we consider the head difference across this confining unit. At the upper surface of the confining unit the head is simply that in the upper aquifer in cell i,j,k--$h_{i,j,k}$. Just below the lower surface of the confining unit, however, unsaturated conditions prevail, so that the pressure sensed on the lower surface of the confining unit is atmospheric--taken as zero in the model formulation. Thus the head at the base of the confining unit is simply the elevation at that point-i.e., the elevation of the top of the lower cell. If this elevation is designated $TOP_{i,j,k+1}$, the flow through the confining bed is obtained by substituting $TOP_{i,j,k+1}$ for $h_{i,j,k+1}$ in equation (55),

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

Thus the flow will be downward, from cell i,j,k to cell i,j,k+1 (i.e., following the convention of equation (26), $q_{i,j,k+1/2}$ will be negative); but under this condition the flow will no longer be dependent on the water level, $h_{i,j,k+1}$, in the lower cell. The simplest approach to this probelm in formulating the equation for cell i,j,k would be to substitute the flow expression of equation (56) into equation (54), in place of the expression given in (55). However, if we consider the matrix of coefficients of the entire system of finite difference equations (matrix [A] of equation (27)), direct substitution of the expression in (56) into the equation for node i,j,k would render this matrix unsymmetric, generating problems in the solution process. To avoid this condition, an alternative approach is used. The flow term of equation (55) is allowed to remain on the left side of equation (54). The flow into cell i,j,k as computed by this term, is

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

(where in this case, since $h_{i,j,k} > h_{i,j,k+1}$, the computed flow is negative, indicating movement out of cell i,j,k.) The "actual" flow into cell i,j,k is given by equation (56) as $CV_{i,j,k+1/2}(TOP_{i,j,k+1} - h^{m}_{i,j,k})$ (where again $h^{m}_{i,j,k} > TOP_{i,j,k+1}$ indicating movement out of the cell). A correction term, q_c, can be obtained by subtracting equation (56) from equation (55), i.e.

$$q_c = (\text{computed flow into cell } i,j,k) - (\text{"actual" flow into all } i,j,k) = CV_{i,j,k+1/2}(h^{m}_{i,j,k+1} - TOP_{i,j,k+1}) \qquad (57)$$

To compensate for allowing the computed flow to remain on the left side of equation (54), the term q_c is added to the right side of equation (54). In the operation of the model, equation (54), which is identical to equation (24), is rearranged to the form of equation (26); and in practice, the term q_c is added to the right side, RHS, of equation (26). This immediately introduces a difficulty, since q_c contains the term $h^{m}_{i,j,k+1}$, and all terms involving unknown heads must be kept on the left side of equation (26). to circumvent this difficulty, q_c is actually computed using the value of $h^{m}_{i,j,k+1}$ from the preceding interation, rather than that from the current iteration, i.e.

$$q_{c,n} = CV_{i,j,k+1/2}(h^{m,n-1}_{i,j,k+1} - TOP_{i,j,k+1}) \qquad (58)$$

where $q_{c,n}$ is the value of q_c to be added to RHS in the n^{th} iteration, and $h^{m,n-1}_{i,j,k+1}$ is the value of $h^{m}_{i,j,k+1}$ from the preceding iteration, n-1. As convergence is approached the difference between $h^{m,n-1}_{i,j,k+1}$ and $h^{m,n}_{i,j,k+1}$ becomes progressively smaller, and the approximation involved in (58) thus becomes

more accurate. In the first iteration of each time step, the initial trial value of $h_{i,j,k+1}$ is used in computing q_c.

The process described above is used in formulating the equations for cell i,j,k when the underlying cell, i,j,k+1, has "dewatered"-i.e., when the water level in i,j,k+1 has fallen below the top of the cell. A correction must also be applied in formulating the equations for the dewatered cell itself. To examine this correction, we now take cell i,j,k to be the dewatered cell, and we consider flow into i,j,k from the overlying cell, i,j,k-1. For this case, the computed flow into cell i,j,k from above is $CV_{i,j,k-1/2}(h^m_{i,j,k-1} - h^m_{i,j,k})$ whereas the "actual" flow into the cell is $CV_{i,j,k-1/2}(h^m_{i,j,k-1} - TOP_{i,j,k})$. The difference, computed minus "actual" flow, is thus $q_c' = CV_{i,j,k-1/2}(TOP_{i,j,k} - h^m_{i,j,k})$ where q_c' should be added to the right hand side of equation (54) or (26). From a programming point of view, the most efficient way to handle this correction is to add the term $CV_{i,j,k-1/2}$ to HCOF on the left side of equation (26), while adding the term $(CV_{i,j,k-1/2} \cdot TOP_{i,j,k})$ to the RHS term. Because HCOF forms part of the coefficient of $h^m_{i,j,k}$, which falls on the main diagonal of the coefficient matrix, this correction does not affect the symmetry of the coefficient matrix; at the same time, the problems entailed in placing an unknown head value on the right side of the equation are avoided.

In summary, whenever dewatering of a cell occurs, two corrections must be made--one in formulating equation (26) as it applies to the overlying cell, and one in formulating equation (26) as it applies to the dewatered

cell itself. These two corrections are discussed separately above, in each case using the designation i,j,k to represent the cell for which equation (26) is formulated. It is important to keep in mind, however, that both corrections are applied in any dewatering event, and that the form of the corrections has been developed to preserve the symmetry of the coefficient matrix [A] of equation (27), and to maximize program efficiency.

In the program described herein, the user specifies whether or not the procedure for limiting vertical flow under dewatered conditions is to be implemented. This is done through the layer type-flag, LAYCON, as discussed in the section on data requirements.

Storage Formulation

In the formulation of storage terms, the program described herein distinguishes between layers in which storage coefficient values remain constant throughout the simulation, and those in which the storage coefficient may "convert" from a confined value to a water table value, or vice-versa, as the water level in a cell falls below or rises above the top of the cell. This distinction is made through the use of the layer flag, LAYCON, as described in the section on data requirements.

For a layer in which storage coefficient is to remain constant during the simulation, the storage formulation is based upon a direct application of the storage expression in equation (24) or (54). This expression, which applies to an individual cell, i,j,k, has the form

$$\frac{\Delta V}{\Delta t} = SS_{i,j,k}\,(\Delta r_j \Delta c_i \Delta v_k)\,\frac{h^{m}_{i,j,k} - h^{m-1}_{i,j,k}}{t_m - t_{m-1}} \qquad (60)$$

where $\frac{\Delta V}{\Delta t}$ is the rate of accumulation of water in the cell, and as such must appear on the right side of equation (24) or (54); $SS_{i,j,k}$ is the specific storage of the material in cell i,j,k; Δr_j, Δc_i and Δv_k are the cell dimensions; $h^m_{i,j,k}$ is the head in cell i,j,k at the end of time step m; $h^{m-1}_{i,j,k}$ is the head in cell i,j,k at the end of time step m-1; t_m is the time at the end of time step m; and t_{m-1} is the time at the end of time step m-1. In equation (26) the notation $SC1_{i,j,k}$ was introduced, where $SC1_{i,j,k} = SS_{i,j,k}\Delta r_j \Delta c_i \Delta v_k$. In this report the term $SC1_{i,j,k}$ is termed the "storage capacity" or the "primary storage capacity" of cell i,j,k; the "primary" designation is used to distinguish $SC1_{i,j,k}$ from a secondary storage capacity which is used when storage term conversion is invoked, as explained in the following section. Using the concept of storage capacity, the expression for rate of accumulation in storage in cell i,j,k can be written

$$SC1_{i,j,k}(h^m_{i,j,k} - h^{m-1}_{i,j,k})/(t_m - t_{m-1}) \quad .$$

This expression is separated into two terms in equation (26), $SC1_{i,j,k}\, h^m_{i,j,k}/(t_m - t_{m-1})$, which is incorporated in the left side of (26) through the term $HCOF_{i,j,k}$, and $SC1_{i,j,k}\, h^{m-1}_{i,j,k}/(t_m - t_{m-1})$, which is included in the term $RHS_{i,j,k}$ on the right side of (26).

The input to the Block-Centered Flow Package requires specification of dimensionless storage coefficient values in each layer of the model; for a confined layer these storage coefficient values are given by the specific storage of the cell material multiplied by layer thickness in the cell, $SS_{i,j,k}\, \Delta v_k$; for an unconfined layer they are equal to the specific yield of the material in the cell. The incorporation of layer thickness into the confined

storage term maintains the flexibility of the program to represent layers of varying thickness, and to implement either the direct three-dimensional or "quasi-three-dimensional" conceptualizations of vertical discretization. The storage coefficient values are read layer by layer; they are designated as array sf1 in the input instructions. These values are then multiplied by the cell areas, $\Delta r_j \Delta c_i$, to create storage capacity values, and they are stored in the SC1 array.

Storage Term Conversion

The primary storage capacity described above, $SC1_{i,j,k}$ is adequate for simulations in which the water level in each individual cell remains either above the top of the cell or below the top of the cell throughout the course of the simulation. If the water level crosses the top of a cell during a simulation--i.e., if the water level in a confined (fully saturated) cell falls below the top of the cell as a result of simulated pumpage, or if the water level in an unconfined cell rises above the top of the cell--then in effect the system "converts" from confined to water table conditions, or vice versa, during the simulation. Where these conditions appear to be possible, the user may invoke storage term conversion for the entire layer through use of the layer-type flag. When this is done, the primary storage capacity, $SC1_{i,j,k}$ for any cell in the layer will represent the confined storage coefficient multiplied by cell area; a secondary storage capacity, $SC2_{i,j,k}$ is used to represent specific yield multiplied by cell area. Values of confined storage coefficient for each cell in the layer are read through the two-dimensional input array sf1. These confined storage

coefficient values are multiplied by cell areas to obtain confined storage capacities, which are stored in the array SC1. Values of specific yield for each cell in the layer are read through the two-dimensional input array sf2. These specific yield values are multiplied by cell areas to obtain unconfined storage capacities, which are stored in array SC2.

In a layer which has been designated for storage term conversion, the expression for rate of accumulation in storage in cell i,j,k is formulated as follows

$$\frac{\Delta V}{\Delta t} = \frac{SCB\ (h^{m}_{i,j,k} - TOP_{i,j,k}) + SCA\ (TOP_{i,j,k} - h^{m-1}_{i,j,k})}{t_m - t_{m-1}} \qquad (61)$$

where again $\frac{\Delta V}{\Delta t}$ is rate of accumulation of water in storage in cell i,j,k and as such must appear on the right side of equation (24) or (54); SCA is the storage capacity in effect in cell i,j,k at the start of the time step; and SCB is the "current" storage capacity--that is, the storage capacity in effect during the iteration in process. Consider a case in which the head in cell i,j,k at the beginning of time step m ($h^{m-1}_{i,j,k}$) is above the top of the cell. Since there is no free surface in the cell at the start of the time step, the storage capacity at that time is taken as the confined storage capacity--that is, SCA is set equal to $SC1_{i,j,k}$. If, during a given iteration for time step m, the computed value of head for the end of the time step ($h^{m}_{i,j,k}$) is found to be above the top of the cell, SCB for the following iteration is also set equal to $SC1_{i,j,k}$; equation (61) for that

iteration then reverts to the form of equation (60). However, if the computed value of $h^{m}_{i,j,k}$ in a given iteration turns out to be below the top of the cell, as shown in figure 30, the value of SCB for the following iteration is set equal to SC2, the unconfined storage capacity. In this case the computed rate of release of water from storage in the time step has two components:

$$SC1_{i,j,k}\ (TOP_{i,j,k} - h^{m-1}_{i,j,k})/(t_m - t_{m-1}) \quad ,$$

the rate of release from confined or compressive storage; and

$$SC2_{i,j,k}\ (h^{m}_{i,j,k} - TOP_{i,j,k})/(t_m - t_{m-1}) \quad ,$$

the rate of release from water table storage.

If the head at the beginning of the time step, $h^{m-1}_{i,j,k}$, is below the top of cell i,j,k, so that a free surface exists within the cell, SCA in equation (61) is set equal to $SC2_{i,j,k}$. If, during an iteration for time step m, the computed value of head for the end of the time step turns out to be below the top of the cell, SCB in the subsequent iteration is also set equal to $SC2_{i,j,k}$ and equation (61) again reverts to the form of equation (60). However, if the computed head for the end of the time step turns out to be above the top of the cell, SCB in the subsequent iteration is set equal to $SC1_{i,j,k}$, the confined storage capacity. This situation occurs during intervals of rising water level, and again two components are computed for the rate of accumulation of water in storage--one corresponding to unconfined or water table storage and one corresponding to confined or compressive storage.

Equation (61) can be rearranged as follows

$$\frac{\Delta V}{\Delta t} = \frac{SCB}{t_m - t_{m-1}}\, h^{m}_{i,j,k} + \frac{SCA\ (TOP_{i,j,k} - h^{m-1}_{i,j,k}) - SCB*TOP_{i,j,k}}{t_m - t_{m-1}} \qquad (62)$$

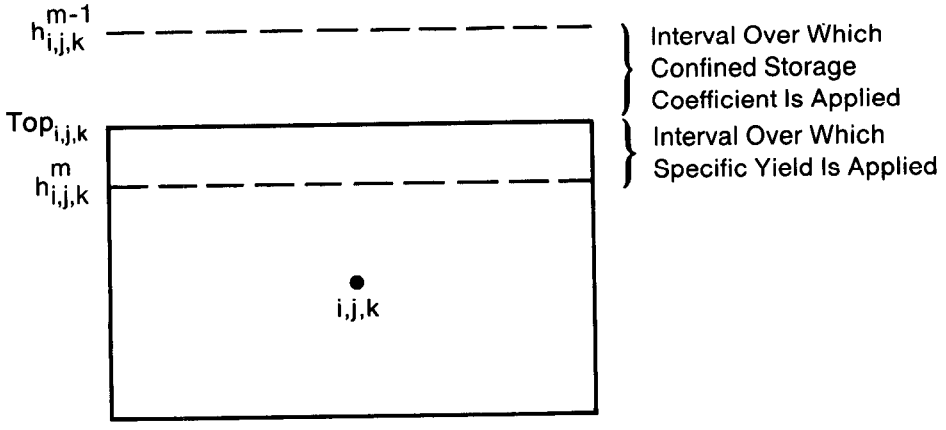


Figure 30.—A model cell which uses two storage factors during one iteration.

Again, $\frac{\Delta V}{\Delta t}$ represents rate of accumulation in storage and as such would appear on the right in equation (24) or (54). In the formulation of equation (26), therefore, the term $\frac{SCB}{t_m - t_{m-1}}$ is subtracted from $HCOF_{i,j,k}$ on the left hand side, while the term $\frac{SCA(TOP_{i,j,k} - h^{m-1}_{i,j,k}) - SCB*TOP_{i,j,k}}{t_m - t_{m-1}}$ is added to $RHS_{i,j,k}$ on the right.

Applicability and Limitations of Optional Formulations

The options for calculation of horizontal conductance under water table conditions, limitation of vertical flow under desaturating conditions, and storage term conversion were all developed on the assumption that each model layer corresponds to a distinct aquifer or permeable horizon, and that these horizons are separated by distinct units of low permeability. Use of these options where these conditions are not satisfied may lead to a variety of problems and inaccuracies in simulation. For example, if the option for horizontal conductance calculation under water table conditions is used where a water table aquifer is represented by several model layers, and the water table is expected to traverse more than one layer during simulation, incorrect (and irreversible) conversion of cells to a no-flow condition may occur as iterations are carried out. Thus care should be exercised in the decision to use any of the three options noted above.

Data Requirements

The fundamental variables controlling cell-to-cell flow and storage in the model are entered through the Block-Centered-Flow Package input. These variables, depending on the options which are invoked, may include

transmissivity, hydraulic conductivity, specific yield, confined storage coefficient, vertical leakance, aquifer bottom elevation and aquifer top elevation. Each of these variables is entered using the utility array-reader module U2DREL, which is described in Chapter 14. This module either reads a two-dimensional array of data for a single layer, or accepts a single value provided by the user and applies that value throughout the array, for all cells in the layer.

The model utilizes a layer-type code to classify layers according to the simulation options that are used. In particular, the layer-type code indicates whether specified transmissivity values are to be used, or transmissivities are to be calculated at each iteration as the product of hydraulic conductivity and saturated thickness; whether storage term conversion is to be used; and whether limitation of vertical flow from above is to be invoked under dewatered conditions. Because the layer-type code identifies the options to be employed in a given layer, it indicates the kinds of data required for the layer, and thus identifies the data arrays to be read. The data are entered layer by layer; for each layer a set of two-dimensional arrays, one array for each required parameter, is read in turn. That is, all of the required arrays for layer 1 are read initially, in sequence, then all of the arrays for layer 2, and so on until all layers have been covered. This method of data organization provides a simpler input process than would be possible using the alternative of a series of three-dimensional arrays corresponding to the various parameters.

Within each layer the required parameters should be specified for every cell, including constant-head and no-flow cells. For no-flow cells,

the entered values are never used in calculation, and thus any values may be specified; for constant head cells, the storage terms are not used but the other parameters are, and realistic values for those parameters must be entered.

Two parameters, transmissivity and hydraulic conductivity, each require the designation of two values at each cell--one in the row direction and one in the column direction. To reduce input effort, only a single array is read for each of these parameters, giving only the values in the row direction; these row-direction values are subsequently multiplied by an anisotropy factor to obtain the corresponding column-direction values. A single value of the anisotropy factor is specified by the user for each layer, through the one-dimensional array TRPY (NLAY).

Vertical leakance terms (Vcont, or $K_z/\Delta z$) are associated with each layer except the lowermost; the values associated with a given layer actually apply to the interval between that layer and the next lower layer. For example, the array of Vcont values entered during the input sequence for layer 1 actually applies to the interval between the midpoint of layer 1 and the midpoint of layer 2.

In addition to the terms mentioned above, the Block-Centered Flow Package input includes cell dimensions (DELR and DELC), a flag to indicate whether the simulation is transient or steady state (ISS), and a flag to indicate whether cell-by-cell flow terms are to be saved (IBCFCB). If the ISS flag is set for steady-state conditions (ISS $\neq$ 0), no space is allocated for storage coefficient or specific yield and storage calculations are skipped. Thus for steady-state runs, arrays of storage coefficients or

specific yields must not be included in the input data; if they are included, the data sequence will be misread. Note that erroneous specification of ISS or of a LAYCON value will also cause misreading of the data array sequence.

Four types of layer are recognized by the model, incoporating various combinations of the options provided by the Block-Centered-Flow Package. These four layer types are identified by their layer-type codes, which are stored in the one-dimensional array LAYCON (NLAY). The code values and the corresponding layer characteristics are given below.

Layer-type 0--In this category there is no provision for modification of transmissivity as water level varies, for storage term conversion, or for limitation of vertical flow from above if water level falls below the top of the cell. This layer type is normally used to simulate confined conditions, but could also be used to simulate a layer in which unconfined conditions will always prevail, provided drawdowns are expected to be a small fraction of layer thickness and flow from the overlying layer (if present) is expected to be negligible. If the simulation is transient, storage coefficient or specific yield values are entered in the input array sf1(NCOL, NROW); then row-direction transmissivities are entered in the input array Tran (NCOL, NROW); and following the transmissivities, unless the layer is the lowermost in the model, vertical leakance values are entered in the input array Vcont (NCOL, NROW). Again, parameter values may be specified by providing the entire array, or by providing a single default value which is applied to all cells of the layer. The parameter values assigned at the beginning of a simulation in this type of layer are retained without change throughout the simulation.

Layer-type 1--This layer type is utilized only in a single-layer model or in the uppermost layer of a model, and only where unconfined conditions are expected to persist in the layer throughout the entire period of simulation. No provision is made for storage term conversion, by virtue of the assumption that water table conditions will always prevail; and no provision is made for limiting flow from above under dewatered conditions, since layer-type 1 is used only for the uppermost layer of a model. However, transmissivities are computed at each iteration as the products of hydraulic conductivity and saturated thickness values within the layer. Thus the input data includes hydraulic conductivity and cell bottom elevation, rather than transmissivity. If the simulation is indicated as transient, specific yield values are entered in the input array sf1(NCOL, NROW). Row direction hydraulic conductivity values are then entered in the input array HY(NCOL, NROW) and cell bottom elevations are entered in the array BOT(NCOL, NROW). If the model contains more than one layer, vertical leakance values are entered in the input array Vcont(NCOL, NROW). Because use of this layer type would be inappropriate except in the uppermost layer, a check of the layer number is made whenever LAYCON is given a value of one; if the layer number is not also equal to one, indicating the uppermost model layer, an error message is printed.

Layer-type 2--This layer type is used where the situation may alternate between confined and unconfined conditions, so that storage term conversion and limitation of flow from above under dewatered conditions are both desirable; but where the saturated thickness is expected to remain everywhere a high fraction of the layer thickness throughout the period of simulation,

so that recalculation of transmissivity as the product of hydraulic conductivity and saturated thickness is not necessary. The storage term conversion option requires that both a confined storage coefficient and a specific yield value be specified for each cell, and that the top elevation be specified for each cell; the top elevation is also used in the option to limit flow from above under dewatered conditions. If the simulation is transient, confined storage coefficient values are entered in the input array sf1(NCOL, NROW). Transmissivity values are then entered in the array Tran(NCOL, NROW). Unless the layer is the lowermost in the model, vertical leakance values are next entered in the array Vcont(NCOL, NROW). Specific yield values are then entered in the array sf2(NCOL, NROW) if the simulation is transient; and finally layer top elevations are entered in the array TOP (NCOL, NROW).

Layer-Type 3--This layer type incorporates all of the Block-Centered-Flow options associated with water table conditions. Transmissivities are recalculated at each iteration using hydraulic conductivities and layer bottom elevations, and both storage term conversion and limitation of flow from above under dewatered conditions are implemented. The required data thus includes hydraulic conductivities, layer bottom elevations, confined storage coefficients (if transient), specific yields (if transient), vertical leakances and layer top elevations. Confined storage coefficients are entered in the input array sf1 (NCOL, NROW); hydraulic conductivity values are then entered in the array HY(NCOL, NROW), and aquifer bottom elevations in BOT(NCOL, NROW). Unless the layer is the lowermost in the model, vertical leakance values are next entered in the array Vcont(NCOL, NROW). Specific yield values are then entered in the array

sf2(NCOL,NROW); and finally aquifer top elevations are entered in the array TOP(NCOL, NROW).

The input sequence is outlined in the following section. Both of the utility modules which are used are described in Chapter 14, and the required formats are illustrated in the "Sample Input to the BCF Package" and in appendix D.

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; the terms which are saved will include cell-by-cell storage terms, cell-by-cell constant head flows, and internal cell-by-cell flows.

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, rather than saved on disk, whenever ICBCFL is set; cell-by-cell storage terms and internal cell-by-cell flows will neither be saved nor printed.

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 alternate between confined and unconfined values. Vertical leakage from above is limited if the layer desaturates.

3 - confined/unconfined--Transmissivity of the layer varies. It is calculated from the saturated thickness and hydraulic conductivity. The storage coefficient may alternate between confined and unconfined values. Vertical leakage from above is limited if the aquifer desaturates.

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 coefficient. Read only for a transient simulation (steady-state flag, ISS, is 0). Note that for Laycon=1, sf1 will always be specific yield, while for Laycon=2 or 3, sf1 will always be confined storage coefficient. For Laycon=0, sf1 would normally be confined storage coefficient; however, layer-type 0 can also be used for simulation of water table conditions where drawdowns are expected to remain everywhere a small fraction of the saturated thickness, and where there is no layer above, or flow from the layer above is negligible; and in this case specific yield values would be entered in sf1.

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 coefficient. 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 coefficient is always specific yield.

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

SAMPLE INPUT TO THE BCF PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{ISS, IBCFCB}	0 0
2	{LAYCON}	1 3 0
3	CONTROL RECORD FOR TRPY ARRAY	31 1. (3F5.0) 6
	TRPY VALUES FOR LAYERS 1,2 AND 3	1. 1. 4.
4	CONTROL RECORD FOR DELR ARRAY	0 1000.
5	CONTROL RECORD FOR DELC ARRAY	31 1000. (10F5.0) 0
	DELC VALUES FOR EACH ROW	2. 2. 1.5 1.5 1.5 1. 1. 1. .7 .5
6	CONTROL RECORD FOR PRIMARY STORAGE FACTOR LAYER 1	0 .2
8	CONTROL RECORD FOR HYDRAULIC CONDUCTIVITY LAYER 1	0 10.
9	CONTROL RECORD FOR BOTTOM LAYER 1	0 -100.
10	CONTROL RECORD FOR VCONT LAYER 1	0 .005
6	CONTROL RECORD FOR PRIMARY STORAGE FACTOR LAYER 2	0 .00007
8	CONTROL RECORD FOR HYDRAULIC CONDUCTIVITY LAYER 2	0 40.
9	CONTROL RECORD FOR BOTTOM LAYER 2	31 1. (15F5.0) 3
	VALUES FOR BOTTOM LAYER 2	-201 -205 -210 -217 -223 -230 -237 -245 -253 -264 -273 -284 -201 -204 -211 -216 -224 -231 -237 -244 -252 -265 -273 -285 -200 -204 -210 -216 -222 -231 -236 -244 -253 -265 -272 -285 -199 -204 -212 -218 -222 -232 -238 -244 -252 -263 -273 -283 -202 -205 -212 -216 -222 -233 -237 -243 -252 -263 -272 -282 -201 -206 -212 -218 -224 -231 -237 -245 -252 -265 -274 -283 -202 -206 -211 -216 -222 -231 -236 -244 -252 -263 -272 -285 -202 -204 -211 -217 -222 -231 -235 -244 -252 -263 -272 -282 -198 -204 -210 -215 -222 -233 -236 -242 -253 -263 -273 -283 -203 -206 -214 -217 -224 -233 -235 -245 -255 -265 -275 -286
10	CONTROL RECORD FOR VCONT LAYER 2	0 .001
11	CONTROL RECORD FOR SECONDARY STORAGE FACTOR LAYER 2	0 .15
12	CONTROL RECORD FOR TOP LAYER 2	32 1. (15F7.2) 0
6	CONTROL RECORD FOR PRIMARY STORAGE FACTOR LAYER 3	0 .00002
7	CONTROL RECORD FOR TRANSMISSIVITY LAYER 3	31 10. (15F6.0) 4
	VALUES FOR TRANSMISSIVITY LAYER 3	50 65 80 95 110 125 140 155 170 185 200 215 70 85 100 115 125 150 160 180 190 203 223 234 85 100 115 132 140 165 176 201 209 220 243 252 101 113 124 139 148 176 190 213 231 239 260 270 122 133 142 158 176 195 210 234 250 260 282 300 137 140 161 175 192 210 224 247 263 275 300 314 152 160 175 190 208 223 242 261 278 282 312 326 162 172 187 201 220 230 251 270 291 295 324 333 170 180 195 209 228 238 259 278 299 303 332 341 175 186 201 215 234 243 264 283 304 308 338 347

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

Module Documentation for the Block-Centered Flow Package

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

Primary Modules

BCF1AL — Allocates space for data arrays.

BCF1RP — Reads all data needed by the package, invokes SBCF1N to reconcile input transmissive values with the IBOUND array, and calculates storage capacities and constant conductances.

BCF1FM — Calculates all coefficients of the system of equations that are not constant and invokes SBCF1H to calculate horizontal-branch conductances in partially saturated layers.

BCF1BD — Calculates flow rates and accumulated flow volumes into and out of storage and constant-head boundaries. When cell-by-cell flow is specified, flow across all sides of each cell is also calculated.

Submodules

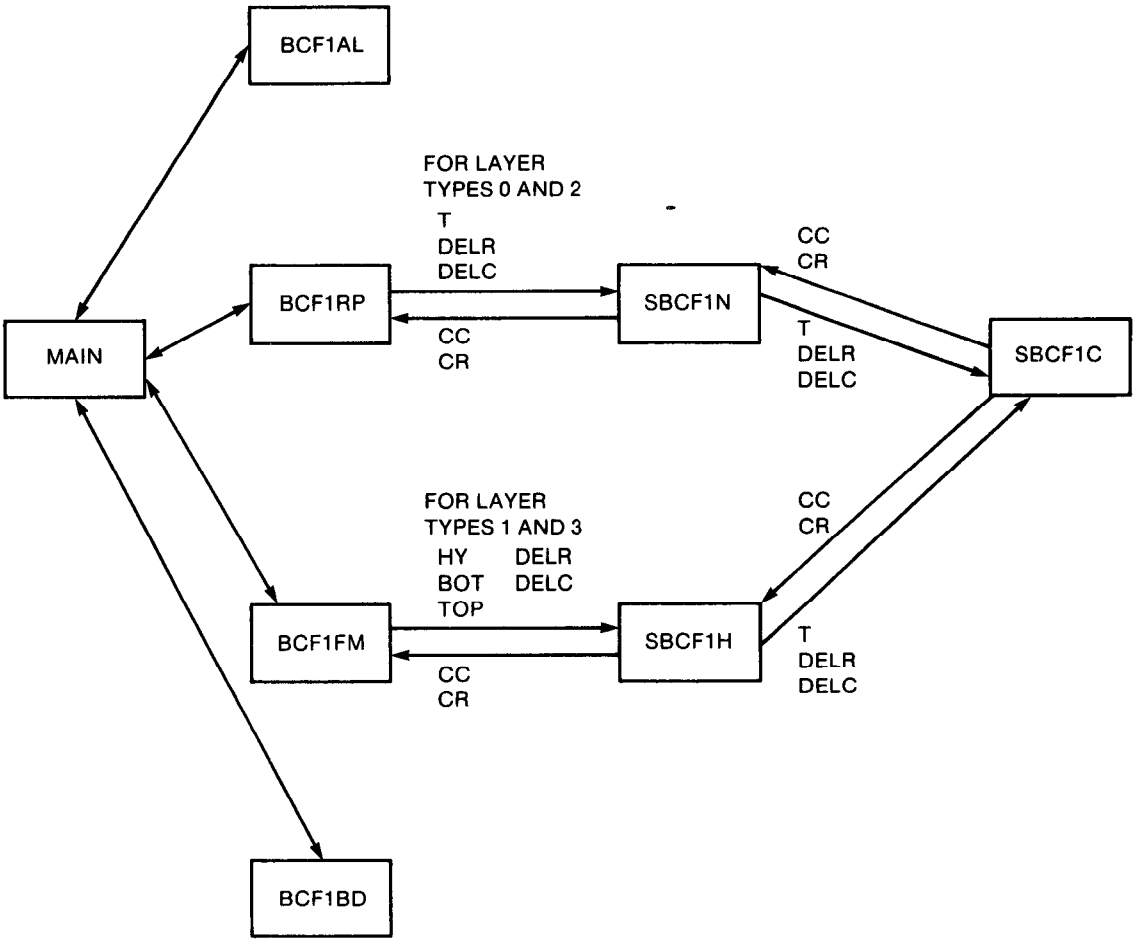
SBCF1N — Reconciles input transmissive values with the IBOUND array and calculates storage capacities and constant conductances. Invokes SBCF1C to calculate horizontal-branch conductances for layers where transmissivity is constant.

SBCF1H — Calculates transmissivity for cells in layers where it depends on heads and invokes SBCF1C to calculate horizontal-branch conductances.

SBCF1C — Calculates horizontal-branch conductance from cell transmissivity.

SBCF1B — Calculates cell-by-cell flow terms across cell faces.

SBCF1F — Calculates flow terms (both cell-by-cell and entries to overall budget) for flow to and from constant-head cells.



Explanation

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

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

Narrative for Module BCF1AL

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

1. Print the message identifying the package.

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

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

   (a) Read the layer-type codes.

   0 = confined

   1 = unconfined

   2 = confined/unconfined but transmissivity is constant

   3 = confined/unconfined but transmissivity depends on head

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

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

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

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

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

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

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

5. Allocate space for the following arrays:

SC1	Primary storage capacity;
SC2	Secondary-storage capacity (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 coefficients are needed.

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

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

7. RETURN.

Flow Chart for Module BCF1AL

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

IBCFCB is a flag and a unit number.

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

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

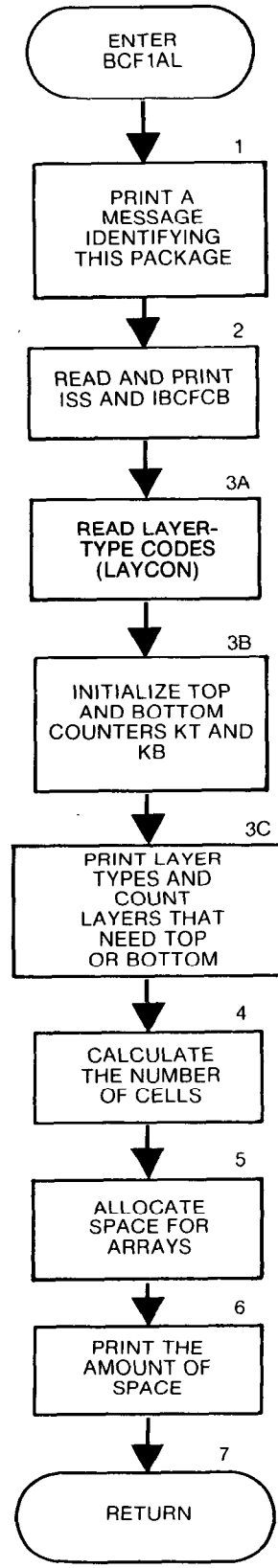
If IBCFCB < 0, flow from constant-head cells will be printed whenever ICBCFL is set.

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

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

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

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



```
      SUBROUTINE BCF1AL(ISUM,LENX,LCSC1,LCHY,LCBOT,
     1     LCTOP,LCSC2,LCTRPY,IN,ISS,NCOL,NROW,NLAY,IOUT,IBCFCB)
C
C-----VERSION 1542 12MAY1987 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', 9/1/87',' 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,I8,' ELEMENTS IN X ARRAY ARE USED BY BCF')
      ISUM1=ISUM-1
      WRITE(IOUT,102) ISUM1,LENX
  102 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      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 U2DREL 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 coefficient.

   (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. For each layer, the vertical conductance to the next lower layer is calculated. Therefore, no vertical conductance is calculated for the lowest layer in the mesh.

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

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

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

4. RETURN.

Flow Chart for Module BCF1RP

DELR is the grid spacing in the row direction.

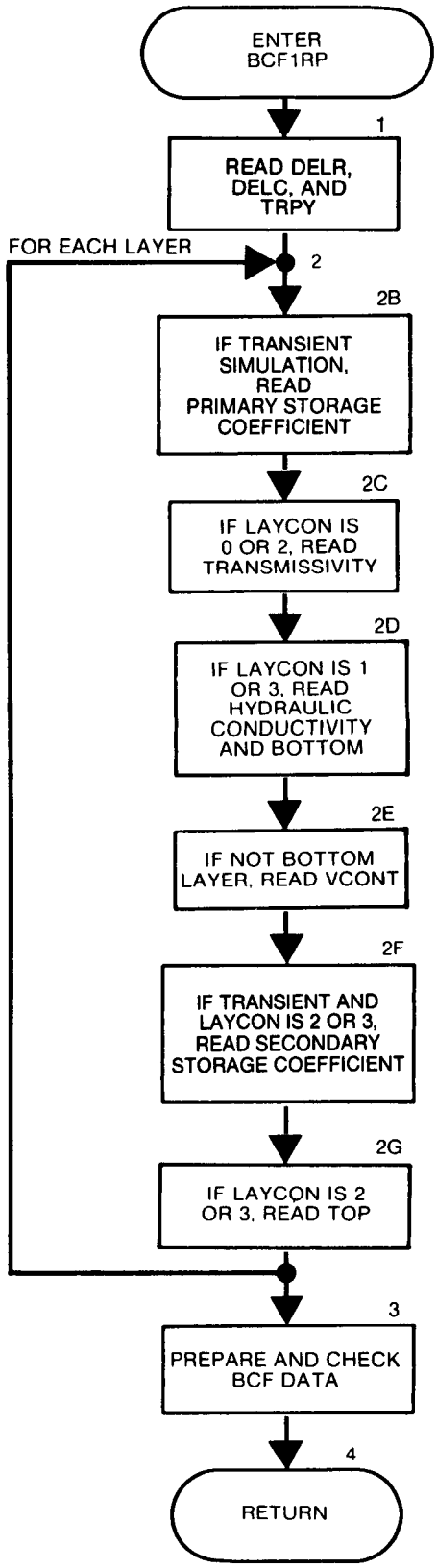
DELC is the grid spacing in the column direction.

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

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

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

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



```
      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 1636 15MAY1987 BCF1RP
C
C     ******************************************************************
C     READ AND INITIALIZE DATA FOR BLOCK-CENTERED FLOW PACKAGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 ANAME
      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) /'    ','PRIM','ARY ','STOR','AGE ','COEF'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1  ANAME(6,2) /'    ','TRAN','SMIS','. AL','ONG ','ROWS'/
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
     1  ANAME(6,3) /'   H','YD. ','COND','. AL','ONG ','ROWS'/
      DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),
     1  ANAME(6,4) /'VERT',' HYD',' CON','D /T','HICK','NESS'/
      DATA ANAME(1,5),ANAME(2,5),ANAME(3,5),ANAME(4,5),ANAME(5,5),
     1  ANAME(6,5) /'    ','    ','    ','    ','  BO','TTOM'/
      DATA ANAME(1,6),ANAME(2,6),ANAME(3,6),ANAME(4,6),ANAME(5,6),
     1  ANAME(6,6) /'    ','    ','    ','    ','    ',' TOP'/
      DATA ANAME(1,7),ANAME(2,7),ANAME(3,7),ANAME(4,7),ANAME(5,7),
     1  ANAME(6,7) /'  SE','COND','ARY ','STOR','AGE ','COEF'/
      DATA ANAME(1,8),ANAME(2,8),ANAME(3,8),ANAME(4,8),ANAME(5,8),
     1  ANAME(6,8) /'COLU','MN T','O RO','W AN','ISOT','ROPY'/
      DATA ANAME(1,9),ANAME(2,9),ANAME(3,9),ANAME(4,9),ANAME(5,9),
     1  ANAME(6,9) /'    ','    ','    ','    ','    ','DELR'/
      DATA ANAME(1,10),ANAME(2,10),ANAME(3,10),ANAME(4,10),ANAME(5,10),
     1  ANAME(6,10) /'    ','    ','    ','    ','    ','DELC'/
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
      KK=K
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,KK,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,KK,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,KK,IN,IOUT)
      CALL U2DREL(BOT(LOCB),ANAME(1,5),NROW,NCOL,KK,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,KK,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,KK,IN,IOUT)
C
C2G-----READ TOP ELEVATION(TOP) IF LAYER TYPE IS 2 OR 3
      CALL U2DREL(TOP(LOCT),ANAME(1,6),NROW,NCOL,KK,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,I+1,K), This array is used to temporarily hold transmissivity.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,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). This array is used to temporarily to hold Vcont.
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).
KK	Module	Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument, which causes problems with some compilers.
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

List of Variables for Module BCF1RP (Continued)

Variable	Range	Definition
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.
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.

†Initially, storage coefficient values are read into these arrays; these values are multiplied by cell areas in submodule SBCF1N to yield storage capacities.

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.

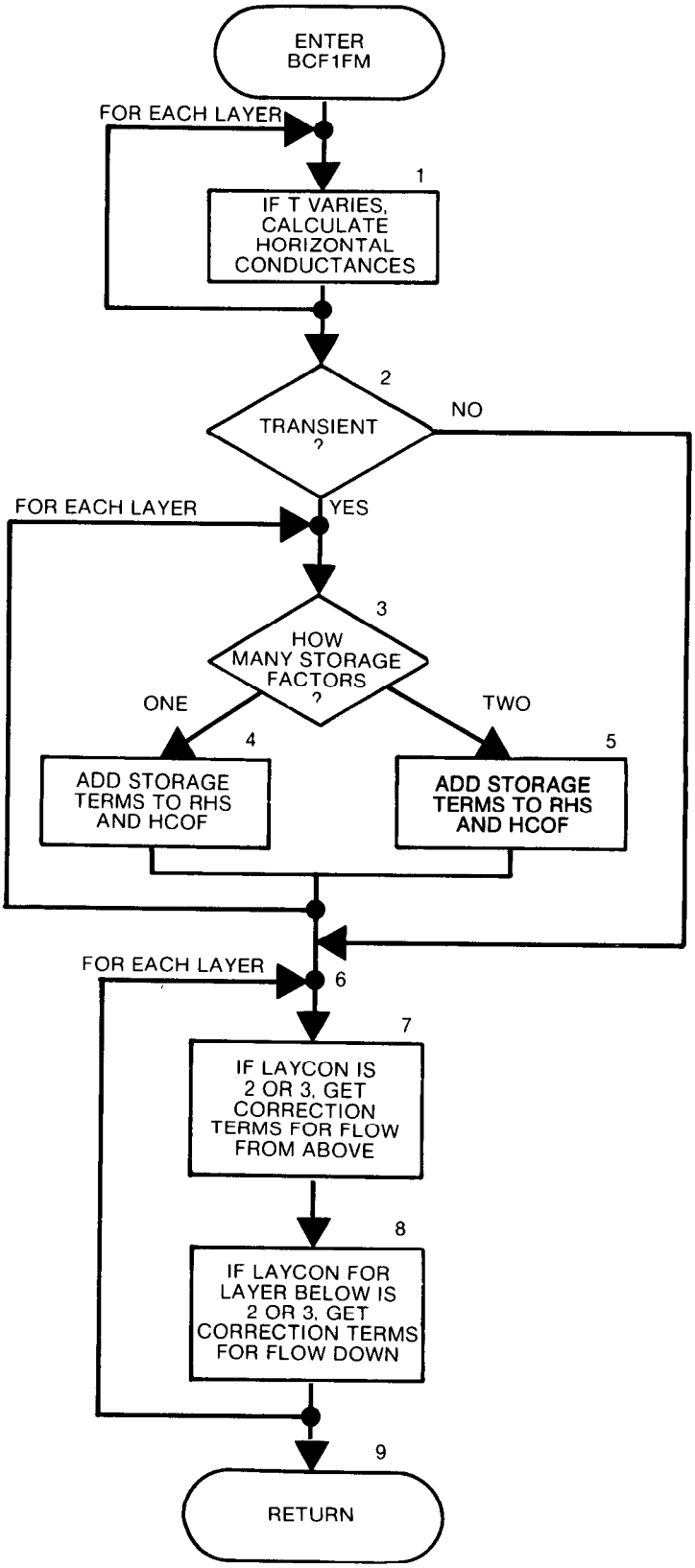
9. RETURN.

†The term storage factor, as used in Subroutine BCF1FM, refers to storage capacity divided by time step length. SOLD is thus equivalent here to $SCA/(t_m-t_{m-1})$, in the notation of equations (61) and (62), while SNEW is equivalent to $SCB/(t_m-t_{m-1})$.

Flow Chart for Module BCF1FM

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

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



```
      SUBROUTINE BCF1FM(HCOF,RHS,HOLD,SC1,HNEW,IBOUND,CR,CC,CV,HY,TRPY,
     1                  BOT,TOP,SC2,DELR,DELC,DELT,ISS,KITER,KSTP,KPER,
     2                  NCOL,NROW,NLAY,IOUT)
C-----VERSION 1640 15MAY1987 BCF1FM
C
C     ******************************************************************
C     ADD LEAKAGE CORRECTION AND STORAGE TO HCOF AND RHS, AND CALCULATE
C     CONDUCTANCE AS REQUIRED
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY),
     1    HOLD(NCOL,NROW,NLAY),SC1(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
     2    IBOUND(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),
     3    CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),
     4    TRPY(NLAY),BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),DELR(NCOL),
     5    DELC(NROW),SC2(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
      KB=0
      KT=0
C
C1------FOR EACH LAYER: IF T VARIES CALCULATE HORIZONTAL CONDUCTANCES
      DO 100 K=1,NLAY
      KK=K
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
C
C1A-----IF LAYER TYPE IS NOT 1 OR 3 THEN SKIP THIS LAYER.
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.1) GO TO 100
      KB=KB+1
C
C1B-----FOR LAYER TYPES 1 & 3 CALL SBCFH1 TO CALCULATE
C1B-----HORIZONTAL CONDUCTANCES.
      CALL SBCF1H(HNEW,IBOUND,CR,CC,CV,HY,TRPY,DELR,DELC,BOT,TOP,
     1         KK,KB,KT,KITER,KSTP,KPER,NCOL,NROW,NLAY,IOUT)
  100 CONTINUE
C
C2------IF THE SIMULATION IS TRANSIENT ADD STORAGE TO HCOF AND RHS
      IF(ISS.NE.0) GO TO 201
      TLED=1./DELT
      KT=0
      DO 200 K=1,NLAY
C
C3------SEE IF THIS LAYER IS CONVERTIBLE OR NON-CONVERTIBLE.
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) GO TO 150
C4------NON-CONVERTIBLE LAYER, SO USE PRIMARY STORAGE
      DO 140 I=1,NROW
      DO 140 J=1,NCOL
      IF(IBOUND(J,I,K).LE.0) GO TO 140
      RHO=SC1(J,I,K)*TLED
      HCOF(J,I,K)=HCOF(J,I,K)-RHO
      RHS(J,I,K)=RHS(J,I,K)-RHO*HOLD(J,I,K)
  140 CONTINUE
      GO TO 200
C
C5------A CONVERTIBLE LAYER, SO CHECK OLD AND NEW HEADS TO DETERMINE
C5------WHEN TO USE PRIMARY AND SECONDARY STORAGE
  150 KT=KT+1
      DO 180 I=1,NROW
      DO 180 J=1,NCOL
C
C5A-----IF THE CELL IS EXTERNAL THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 180
      TP=TOP(J,I,KT)
      RHO2=SC2(J,I,KT)*TLED
```

```
      RHO1=SC1(J,I,K)*TLED
C
C5B-----FIND STORAGE FACTOR AT START OF TIME STEP.
      SOLD=RHO2
      IF(HOLD(J,I,K).GT.TP) SOLD=RHO1
C
C5C-----FIND STORAGE FACTOR AT END OF TIME STEP.
      HTMP=HNEW(J,I,K)
      SNEW=RHO2
      IF(HTMP.GT.TP) SNEW=RHO1
C
C5D-----ADD STORAGE TERMS TO RHS AND HCOF.
      HCOF(J,I,K)=HCOF(J,I,K)-SNEW
      RHS(J,I,K)=RHS(J,I,K) - SOLD*(HOLD(J,I,K)-TP) - SNEW*TP
C
  180 CONTINUE
C
  200 CONTINUE
C
C6------FOR EACH LAYER DETERMINE IF CORRECTION TERMS ARE NEEDED FOR
C6------FLOW DOWN INTO PARTIALLY SATURATED LAYERS.
  201 KT=0
      DO 300 K=1,NLAY
C
C7------SEE IF CORRECTION IS NEEDED FOR LEAKAGE FROM ABOVE.
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 250
      KT=KT+1
      IF(K.EQ.1) GO TO 250
C
C7A-----FOR EACH CELL MAKE THE CORRECTION IF NEEDED.
      DO 220 I=1,NROW
      DO 220 J=1,NCOL
C
C7B-----IF THE CELL IS EXTERNAL(IBOUND<=0) THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 220
      HTMP=HNEW(J,I,K)
C
C7C-----IF HEAD IS ABOVE TOP THEN CORRECTION NOT NEEDED
      IF(HTMP.GE.TOP(J,I,KT)) GO TO 220
C
C7D-----WITH HEAD BELOW TOP ADD CORRECTION TERMS TO RHS AND HCOF.
      RHS(J,I,K)=RHS(J,I,K) + CV(J,I,K-1)*TOP(J,I,KT)
      HCOF(J,I,K)=HCOF(J,I,K) + CV(J,I,K-1)
  220 CONTINUE
C
C8------SEE IF THIS LAYER MAY NEED CORRECTION FOR LEAKAGE TO BELOW.
  250 IF(K.EQ.NLAY) GO TO 300
      IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 300
      KTT=KT+1
C
C8A-----FOR EACH CELL MAKE THE CORRECTION IF NEEDED.
      DO 280 I=1,NROW
      DO 280 J=1,NCOL
C
C8B-----IF CELL IS EXTERNAL (IBOUND<=0) THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 280
C
C8C-----IF HEAD IN THE LOWER CELL IS LESS THAN TOP ADD CORRECTION
C8C-----TERM TO RHS.
      HTMP=HNEW(J,I,K+1)
      IF(HTMP.LT.TOP(J,I,KTT)) RHS(J,I,K)=RHS(J,I,K)
     1                         - CV(J,I,K)*(TOP(J,I,KTT)-HTMP)
  280 CONTINUE
  300 CONTINUE
C
C9------RETURN
      RETURN
      END
```

List of Variables for Module BCF1FM

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,I+1,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,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.
KK	Module	Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument, which causes problems with some compilers.
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 factor for convertible layers.
†RHO2	Module	Unconfined storage factor 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 factor at the end of the time step for convertible layers.
†SOLD	Module	Storage factor 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.

†Storage factor, as used in Subroutine BCF1FM, refers to storage capacity divided by time step length.

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 ≠0) and has specified a unit number (IBCFCB) for cell-by-cell flow terms for the BCF Package, set the cell-by-cell flag (IBD).

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

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

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

6. Calculate flow from storage in the cell.

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

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

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

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

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

12. RETURN.

Flow Chart for Module BCF1BD

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

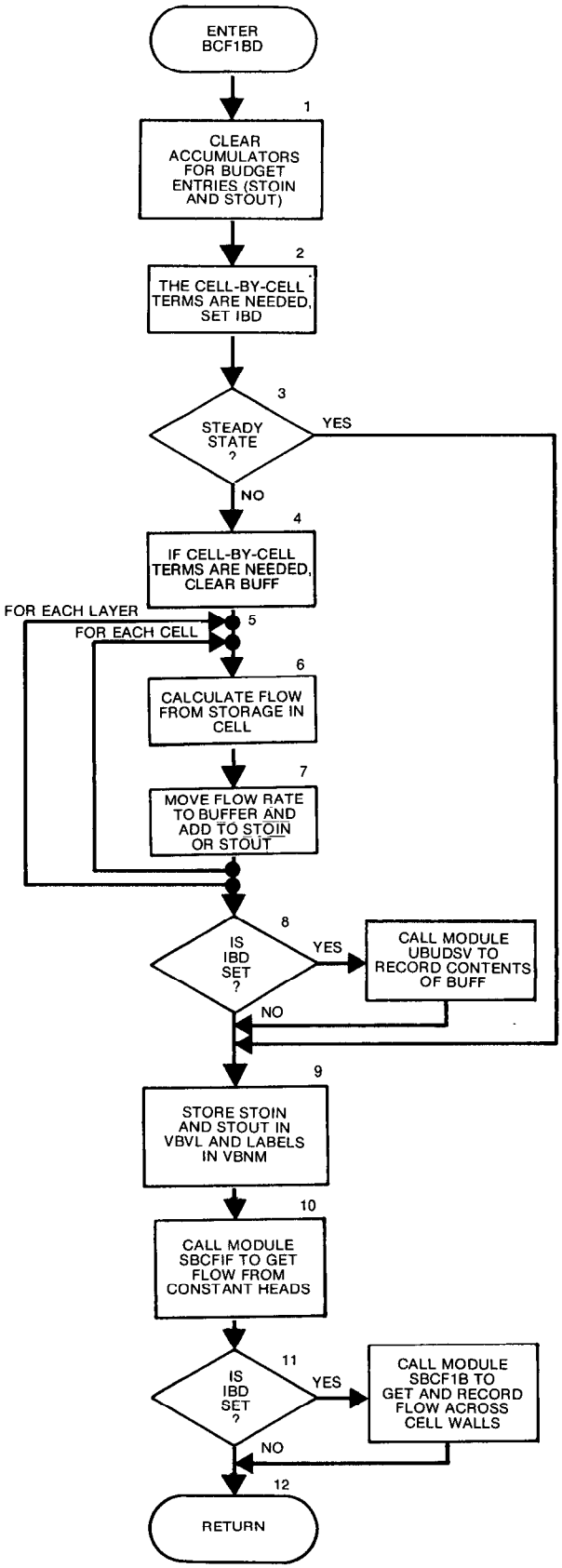
STOUT is an accumulator for flow terms having a negative sign (flow into storage and out of the flow system) for inclusion in the volumetric budget.

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

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

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

VBNM is a table of labels for budget terms.



```
      SUBROUTINE BCF1BD(VBNM,VBVL,MSUM,HNEW,IBOUND,HOLD,SC1,CR,CC,CV,
     1   TOP,SC2,DELT,ISS,NCOL,NROW,NLAY,KSTP,KPER,IBCFCB,
     2   ICBCFL,BUFF,IOUT)
C-----VERSION 1546 12MAY1987 BCF1BD
C
C     ******************************************************************
C     COMPUTE BUDGET FLOW TERMS FOR BCF -- STORAGE, CONSTANT HEAD, AND
C     FLOW ACROSS CELL WALLS
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      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 CAPACITY OR TWO
      IF(LC.NE.3 .AND. LC.NE.2) GO TO 285
C
C6B----TWO STORAGE CAPACITIES
      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 CAPACITY
  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,I+1,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,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 capacity.
SCFA	Module	Temporary name for the primary storage capacity.
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 capacity at the end of the time step.
SOLD*	Module	Storage capacity 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 capacity.
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.

*Note that the variables SOLD and SNEW have different meanings in this subroutine than in BCF1FM.

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 coefficient by DELR and DELC to get the primary storage capacity (SC1).

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

7. RETURN.

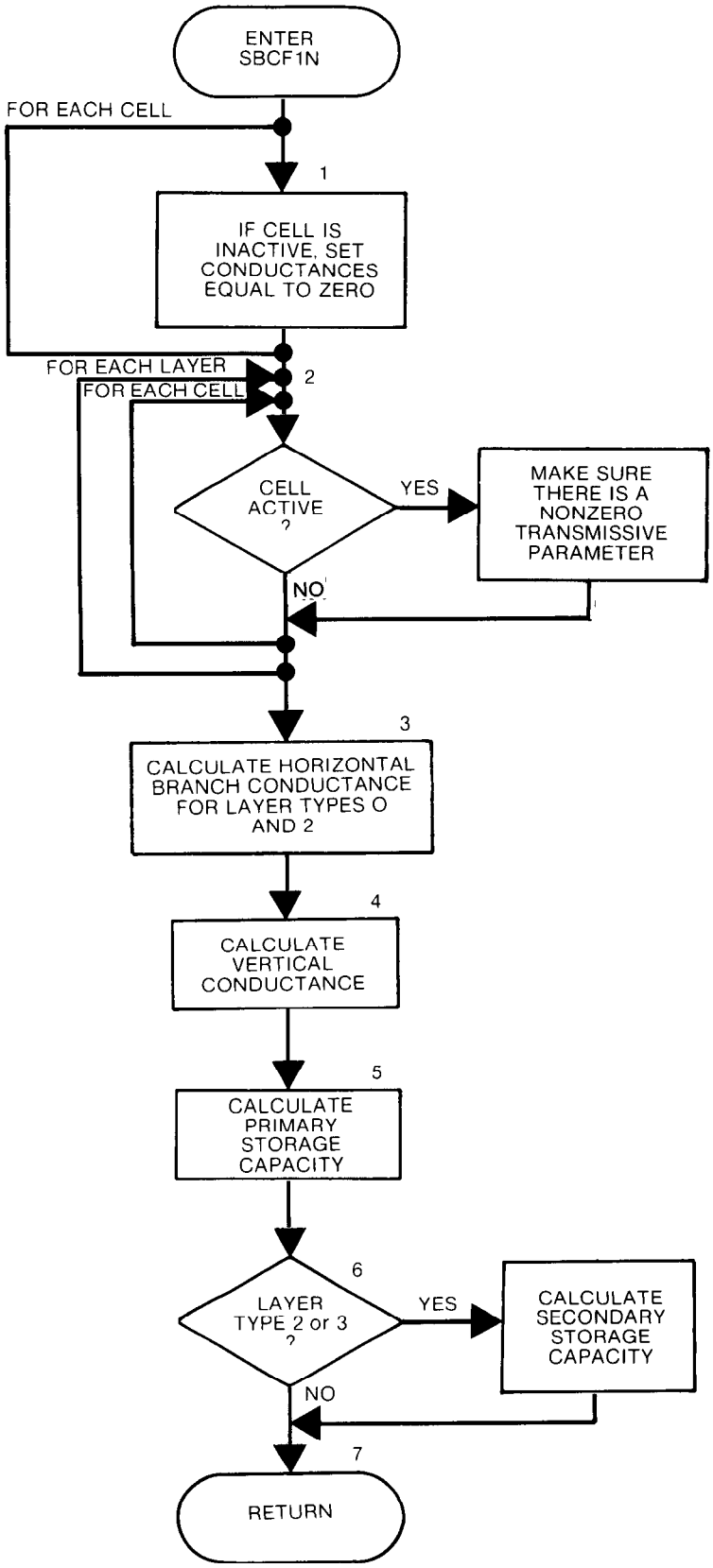
Flow Chart for Module SBCF1N

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

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

Primary storage capacity is taken as specific yield times cell area for unconfined layers, and as confined storage coefficient times cell area for confined or confined/unconfined layers.

Secondary storage capacity is defined for confined/unconfined aquifers and is always taken as specific yield times cell area.



```
      SUBROUTINE SBCF1N(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,
     1          ISS,NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1642 15MAY1987 SBCF1N
C
C     ******************************************************************
C     INITIALIZE AND CHECK BCF DATA
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DOUBLE PRECISION HNEW,HCNV
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
     1    ,SC1(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY)
     2    ,CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY)
     3    ,HY(NCOL,NROW,NLAY),TRPY(NLAY),DELR(NCOL),DELC(NROW)
     4    ,SC2(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------IF IBOUND=0, SET CV=0., CC=0., AND HY=0.
      KB=0
      DO 30 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) KB=KB+1
      DO 30 I=1,NROW
      DO 30 J=1,NCOL
      IF(IBOUND(J,I,K).NE.0) GO TO 30
      IF(K.NE.NLAY) CV(J,I,K)=0.
      IF(K.NE.1) CV(J,I,K-1)=0.
      CC(J,I,K)=0.
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) HY(J,I,KB)=0.
   30 CONTINUE
C
C2------INSURE THAT EACH ACTIVE CELL HAS AT LEAST ONE NON-ZERO
C2------TRANSMISSIVE PARAMETER. IF NOT, CONVERT CELL TO NOFLOW.
      HCNV=888.88
      KB=0
      DO 60 K=1,NLAY
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) GO TO 55
C2A-----WHEN LAYER TYPE 0 OR 2, TRANSMISSIVITY OR CV MUST BE NONZERO
      DO 54 I=1,NROW
      DO 54 J=1,NCOL
      IF(IBOUND(J,I,K).EQ.0) GO TO 54
      IF(CC(J,I,K).NE.0.) GO TO 54
      IF(K.EQ.NLAY) GO TO 51
      IF(CV(J,I,K).NE.0.) GO TO 54
   51 IF(K.EQ.1) GO TO 53
      IF(CV(J,I,K-1).NE.0.) GO TO 54
   53 IBOUND(J,I,K)=0
      HNEW(J,I,K)=HCNV
      WRITE(IOUT,52) K,I,J
   52 FORMAT(1X,'NODE (LAYER,ROW,COL)',3I4,
     1      ' ELIMINATED BECAUSE ALL CONDUCTANCES TO NODE ARE 0')
   54 CONTINUE
      GO TO 60
```

```
C
C2B-----WHEN LAYER TYPE IS 1 OR 3, HY OR CV MUST BE NONZERO
   55 KB=KB+1
      DO 59 I=1,NROW
      DO 59 J=1,NCOL
      IF(IBOUND(J,I,K).EQ.0) GO TO 59
      IF(HY(J,I,KB).NE.0.) GO TO 59
      IF(K.EQ.NLAY) GO TO 56
      IF(CV(J,I,K).NE.0.) GO TO 59
   56 IF(K.EQ.1) GO TO 57
      IF(CV(J,I,K-1).NE.0.) GO TO 59
   57 IBOUND(J,I,K)=0
      HNEW(J,I,K)=HCNV
      CC(J,I,K)=0.
      WRITE(IOUT,52) K,I,J
   59 CONTINUE
   60 CONTINUE
C
C3------CALCULATE HOR. CONDUCTANCE(CR AND CC) FOR CONSTANT T LAYERS
      DO 65 K=1,NLAY
      KK=K
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) GO TO 65
      CALL SBCF1C(CR,CC,TRPY,DELR,DELC,KK,NCOL,NROW,NLAY)
   65 CONTINUE
C
C4------MULTIPLY VERTICAL LEAKANCE BY AREA TO MAKE CONDUCTANCE
      IF(NLAY.EQ.1) GO TO 69
      K1=NLAY-1
      DO 68 K=1,K1
      DO 68 I=1,NROW
      DO 68 J=1,NCOL
      CV(J,I,K)=CV(J,I,K)*DELR(J)*DELC(I)
   68 CONTINUE
C
C5------IF TRANSIENT MULTIPLY PRIMARY STORAGE COEFFICIENT BY DELR &
C5------DELC TO GET PRIMARY STORAGE CAPACITY(SC1).
   69 IF(ISS.NE.0) GO TO 100
      KT=0
      DO 80 K=1,NLAY
      DO 70 I=1,NROW
      DO 70 J=1,NCOL
      SC1(J,I,K)=SC1(J,I,K)*DELR(J)*DELC(I)
   70 CONTINUE
C
C6------IF LAYER IS CONF/UNCONF MULTIPLY SECONDARY STORAGE COEFFICIENT
C6------BY DELR AND DELC TO GET SECONDARY STORAGE CAPACITY(SC2).
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 80
      KT=KT+1
      DO 75 I=1,NROW
      DO 75 J=1,NCOL
      SC2(J,I,KT)=SC2(J,I,KT)*DELR(J)*DELC(I)
   75 CONTINUE
C
   80 CONTINUE
C
C7------RETURN
  100 RETURN
      END
```

List of Variables for Module SBCF1N

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,I+1,K). This array is used to temporarily hold transmissivity.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) AND (J+1,I,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). This array is used to temporarily hold Vcont.
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.
KK	Module	Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument, which causes problems with some compilers.
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	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the 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.)
TRPY	Package	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.

Narrative for Module SBCF1H

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

1. For each cell, calculate the transmissivity. DO STEPS 2-6.

2. If the cell is inactive, set the transmissivity equal to zero and move on to the next cell.

3. Calculate the thickness of the saturation. In a strictly unconfined layer, the thickness is the head (HNEW) minus the bottom (BOTTOM). In a confined/unconfined layer, the thickness is the head (HNEW) minus the bottom or the top (TOP) minus the bottom, whichever is greater.

4. Check to see if the saturated thickness is greater than zero.

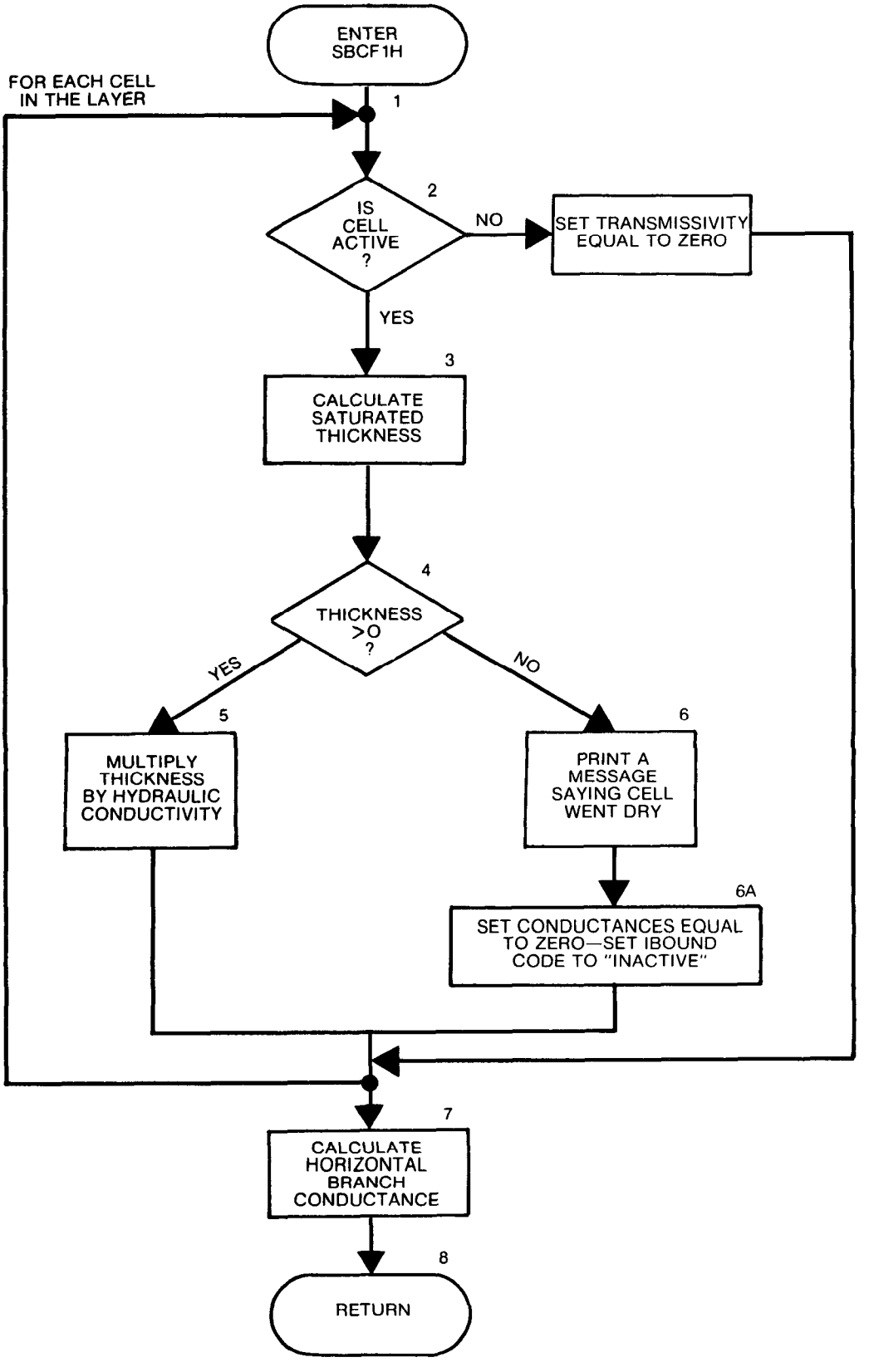
5. If the thickness is greater than zero, the transmissivity of the cell is the thickness times the hydraulic conductivity.

6. If the saturated thickness is less than zero, the cell is dry. Print a message to that effect, set all branch conductances equal to zero, and set the boundary indicator (IBOUND) equal to zero.

7. Call submodule SBCF1C to calculate the horizontal-branch conductances for the layer.

8. RETURN.

Flow Chart for Module SBCF1H



```
      SUBROUTINE SBCF1H(HNEW,IBOUND,CR,CC,CV,HY,TRPY,DELR,DELC
     1,BOT,TOP,K,KB,KT,KITER,KSTP,KPER,NCOL,NROW,NLAY,IOUT)
C
C-----VERSION 1442 31DEC1986 SBCF1H
C
C     ******************************************************************
C     COMPUTE CONDUCTANCE FROM SATURATED THICKNESS AND HYDRAULIC
C     CONDUCTIVITY
C     ******************************************************************
C
C      SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
     1, CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY), CV(NCOL,NROW,NLAY)
     2,HY(NCOL,NROW,NLAY), TRPY(NLAY), DELR(NCOL), DELC(NROW)
     3, BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------CALCULATE TRANSMISSIVITY AT EACH ACTIVE CELL. TRANSMISSIVITY
C1------WILL BE STORED TEMPORARILY IN THE CC ARRAY.
      DO 200 I=1,NROW
      DO 200 J=1,NCOL
C
C2------IF CELL IS INACTIVE THEN SET T=0 & MOVE ON TO NEXT CELL.
      IF(IBOUND(J,I,K).NE.0) GO TO 10
      CC(J,I,K)=0.
      GO TO 200
C
C3------CALCULATE SATURATED THICKNESS.
   10 HD=HNEW(J,I,K)
      IF(LAYCON(K).EQ.1) GO TO 50
      IF(HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
   50 THCK=HD-BOT(J,I,KB)
C
C4------CHECK TO SEE IF SATURATED THICKNESS IS GREATER THAN ZERO.
      IF(THCK.LE.0.) GO TO 100
C
C5------IF SATURATED THICKNESS>0 THEN T=K*THICKNESS.
      CC(J,I,K)=THCK*HY(J,I,KB)
      GO TO 200
C
C6------WHEN SATURATED THICKNESS < 0, PRINT A MESSAGE AND SET
C6------TRANSMISSIVITY, IBOUND, AND VERTICAL CONDUCTANCE =0
  100 WRITE(IOUT,150) K,I,J,KITER,KSTP,KPER
  150 FORMAT(1H0,10('*'),'NODE',3I4,' (LAYER,ROW,COL) WENT DRY'
     1      ,' AT ITERATION =',I3,'  TIME STEP =',I3
     2      ,'  STRESS PERIOD =',I3)
      HNEW(J,I,K)=1.E30
      CC(J,I,K)=0.
      IBOUND(J,I,K)=0
      IF(K.LT.NLAY) CV(J,I,K)=0.
      IF(K.GT.1) CV(J,I,K-1)=0.
      GO TO 200
  200 CONTINUE
C
C7------COMPUTE HORIZONTAL BRANCH CONDUCTANCES FROM TRANSMISSIVITY
      CALL SBCF1C(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
C
C8------RETURN
      RETURN
      END
```

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,I+1,K). This array is used to temporarily hold transmissivity.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,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.

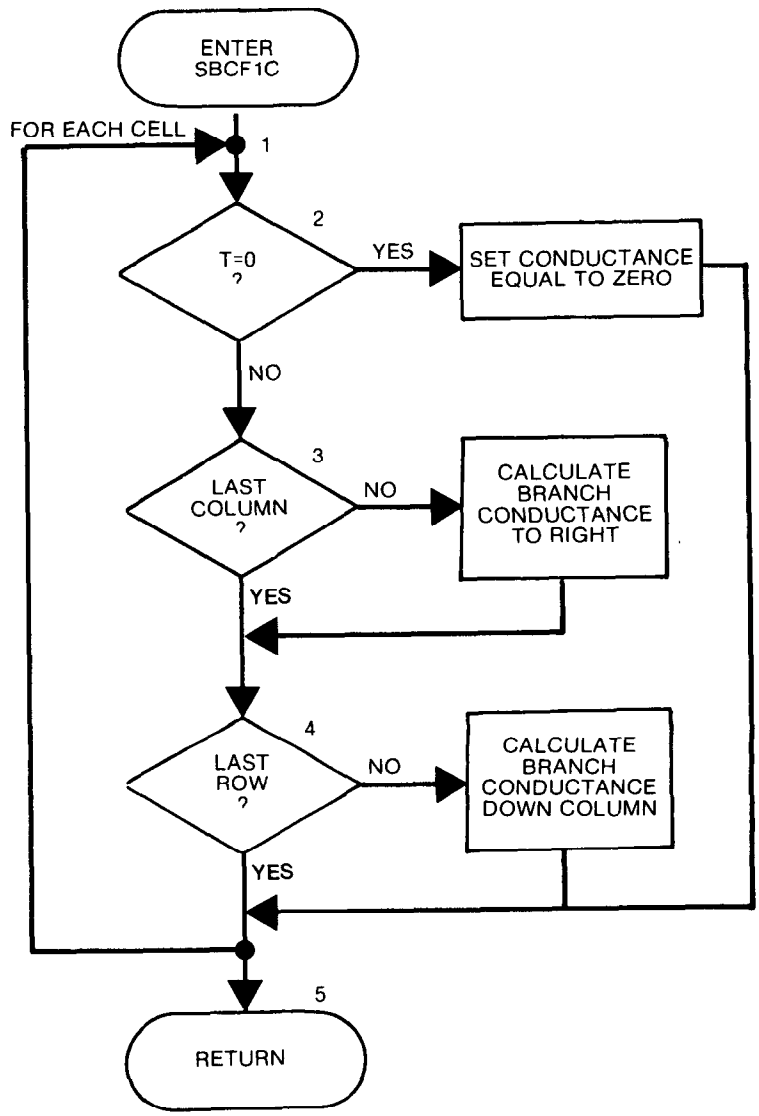
4. If the transmissivity of the cell is not zero and there is a cell in front, calculate the conductance along the column.

5. RETURN.

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

CR(J,I,K) contains the conductance $CR_{i,j+1/2,k}$ between node J,I,K and node J+1,I,K.

Flow Chart for Module SBCF1C



```
      SUBROUTINE SBCF1C(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
C
C
C-----VERSION 1334 22AUG1987 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 ONE 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,I+1,K). This array is used to temporarily hold transmissivity.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
I	Module	Index for rows.
J	Module	Index for columns.
K	Module	Index for layers.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
TRPY	Package	DIMENSION (NLAY), Ratio of transmissivity in the column direction to transmissivity in the row direction.
T1	Module	Temporary field for CC(J,I,K).
T2	Module	Temporary field for CC(J+1,I,K).
YX	Module	TRPY(K)*2.

Narrative for Module SBCF1B

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

1. Clear the buffer (BUFF) in which cell-by-cell flow terms are gathered as they are calculated.

2. For each cell, calculate the flow in the row direction through the right face of the cell and store it in the buffer.

3. Call utility module UBUDSV to write the contents of the buffer.

4. Clear the buffer (BUFF) in which cell-by-cell flow terms are gathered as they are calculated.

5. For each cell, calculate the flow in the column direction through the front face of the cell and store it in the buffer.

6. Call utility module UBUDSV to write the contents of the buffer.

7. Clear the buffer (BUFF) in which cell-by-cell flow terms are gathered as they are calculated.

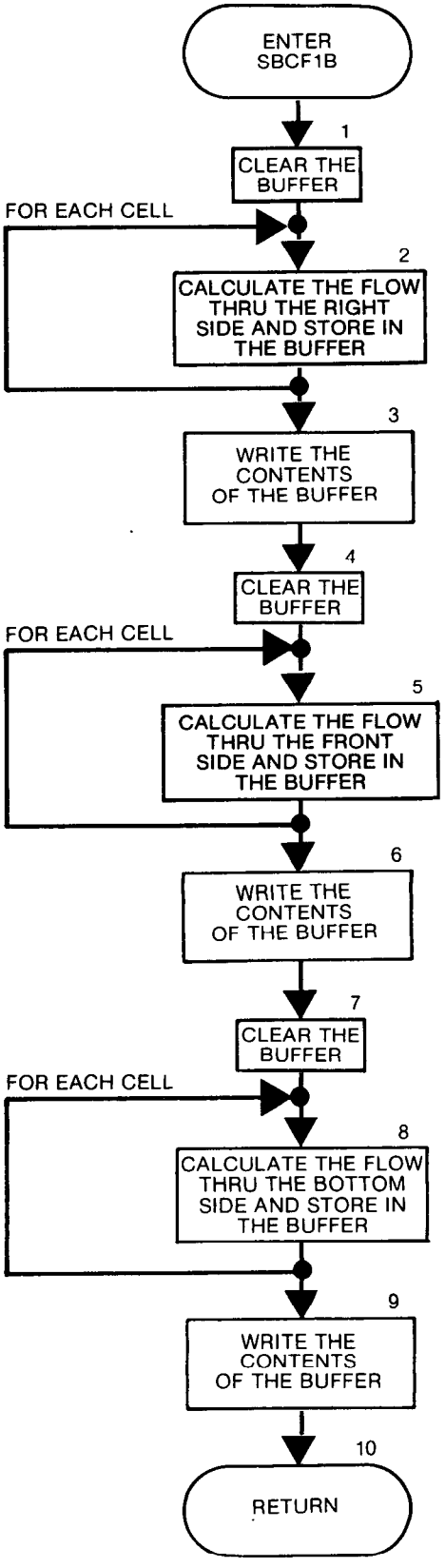
8. For each cell, calculate the flow in the vertical direction through the lower face of the cell and store it in the buffer.

9. Call utility module UBUDSV to write the contents of the buffer.

10. RETURN.

Flow Chart for Module SBCF1B

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



```
      SUBROUTINE SBCF1B(HNEW,IBOUND,CR,CC,CV,TOP,NCOL,NROW,NLAY,
     1      KSTP,KPER,IBCFCB,BUFF,IOUT)
C
C-----VERSION 1548 12MAY1987 SBCF1B
C
C     ******************************************************************
C     COMPUTE FLOW ACROSS EACH CELL WALL
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 TEXT
      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,I+1,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,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.

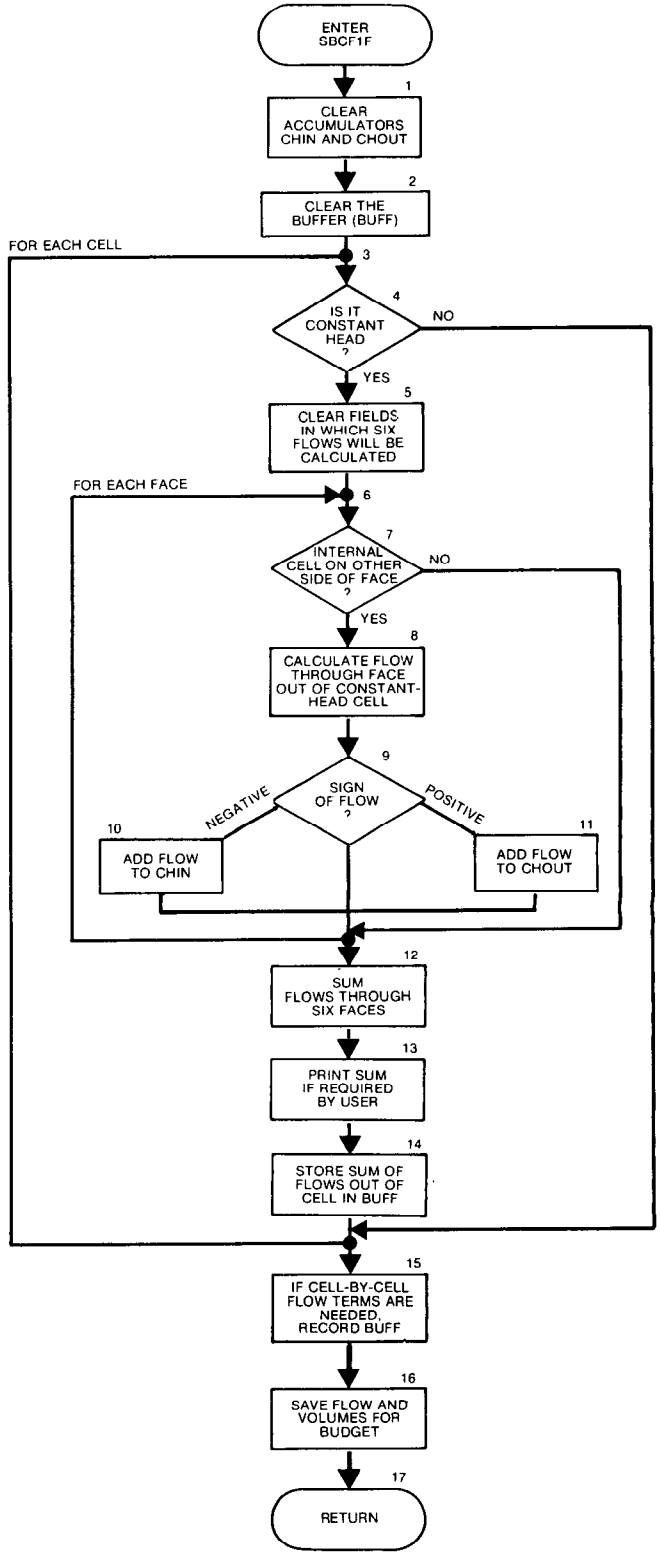
17. RETURN.

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

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

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

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



```
      SUBROUTINE SBCF1F(VBNM,VBVL,MSUM,HNEW,IBOUND,CR,CC,CV,
     1   TOP,DELT,NCOL,NROW,NLAY,KSTP,KPER,IBD,IBCFCB,ICBCFL,
     2   BUFF,IOUT)
C-----VERSION 1549 12MAY1987 SBCF1F
C
C     ******************************************************************
C     COMPUTE FLOW FROM CONSTANT HEAD NODES
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      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), VBNM(4,20), VBVL(4,20),
     3     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) /'   C','ONST','ANT ','HEAD'/
C     ------------------------------------------------------------------
C
C1------CLEAR BUDGET ACCUMULATORS
      CHIN=0.
      CHOUT=0.
C
C2------CLEAR BUFFER IF CELL-BY-CELL FLOW TERM FLAG(IBD) IS SET
      IF(IBD.EQ.0) GO TO 8
      DO 5 K=1,NLAY
      DO 5 I=1,NROW
      DO 5 J=1,NCOL
      BUFF(J,I,K)=0.
    5 CONTINUE
C
C3------FOR EACH CELL IF IT IS CONSTANT HEAD COMPUTE FLOW ACROSS 6
C3-----FACES.
    8 KT=0
      DO 200 K=1,NLAY
      LC=LAYCON(K)
      IF(LC.EQ.3 .OR. LC.EQ.2) KT=KT+1
      DO 200 I=1,NROW
      DO 200 J=1,NCOL
C
C4-----IF CELL IS NOT CONSTANT HEAD SKIP IT & GO ON TO NEXT CELL.
      IF (IBOUND(J,I,K).GE.0)GO TO 200
C
C5-----CLEAR FIELDS FOR SIX FLOW RATES.
      X1=0.
      X2=0.
      X3=0.
      X4=0.
      X5=0.
      X6=0.
C6-----FOR EACH FACE OF THE CELL CALCULATE FLOW THROUGH THAT FACE
C6-----OUT OF THE CONSTANT HEAD CELL AND INTO THE FLOW DOMAIN.
C6-----COMMENTS 7-11 APPEAR ONLY IN THE SECTION HEADED BY COMMENT 6A
C6-----BUT THEY APPLY IN A SIMILAR MANNER TO THE SECTIONS HEADED
C6-----BY COMMENTS 6B-6F.
C
```

```
C6A----CALCULATE FLOW THROUGH THE LEFT FACE
C
C7-----IF THERE IS NOT A VARIABLE HEAD CELL ON THE OTHER SIDE OF THIS
C7-----FACE THEN GO ON TO THE NEXT FACE.
      IF(J.EQ.1) GO TO 30
      IF(IBOUND(J-1,I,K).LE.0)GO TO 30
      HDIFF=HNEW(J,I,K)-HNEW(J-1,I,K)
C
C8-----CALCULATE FLOW THROUGH THIS FACE INTO THE ADJACENT CELL.
      X1=HDIFF*CR(J-1,I,K)
C
C9-----TEST TO SEE IF FLOW IS POSITIVE OR NEGATIVE
      IF (X1) 10,30,20
C
C10----IF NEGATIVE ADD TO CHOUT(FLOW OUT OF DOMAIN TO CONSTANT HEAD).
   10 CHOUT=CHOUT-X1
      GO TO 30
C
C11----IF POSITIVE ADD TO CHIN(FLOW INTO DOMAIN FROM CONSTANT HEAD).
   20 CHIN=CHIN+X1
C
C6B----CALCULATE FLOW THROUGH THE RIGHT FACE
   30 IF(J.EQ.NCOL) GO TO 60
      IF(IBOUND(J+1,I,K).LE.0) GO TO 60
      HDIFF=HNEW(J,I,K)-HNEW(J+1,I,K)
      X2=HDIFF*CR(J,I,K)
      IF(X2)40,60,50
   40 CHOUT=CHOUT-X2
      GO TO 60
   50 CHIN=CHIN+X2
C
C6C----CALCULATE FLOW THROUGH THE BACK FACE.
   60 IF(I.EQ.1) GO TO 90
      IF (IBOUND(J,I-1,K).LE.0) GO TO 90
      HDIFF=HNEW(J,I,K)-HNEW(J,I-1,K)
      X3=HDIFF*CC(J,I-1,K)
      IF(X3) 70,90,80
   70 CHOUT=CHOUT-X3
      GO TO 90
   80 CHIN=CHIN+X3
C
C6D----CALCULATE FLOW THROUGH THE FRONT FACE.
   90 IF(I.EQ.NROW) GO TO 120
      IF(IBOUND(J,I+1,K).LE.0) GO TO 120
      HDIFF=HNEW(J,I,K)-HNEW(J,I+1,K)
      X4=HDIFF*CC(J,I,K)
      IF (X4) 100,120,110
  100 CHOUT=CHOUT-X4
      GO TO 120
  110 CHIN=CHIN+X4
C
C6E----CALCULATE FLOW THROUGH THE UPPER FACE
  120 IF(K.EQ.1) GO TO 150
      IF (IBOUND(J,I,K-1).LE.0) GO TO 150
      HD=HNEW(J,I,K)
      IF(LC.NE.3 .AND. LC.NE.2) GO TO 122
      TMP=HD
      IF(TMP.LT.TOP(J,I,KT)) HD=TOP(J,I,KT)
  122 HDIFF=HD-HNEW(J,I,K-1)
      X5=HDIFF*CV(J,I,K-1)
      IF(X5) 130,150,140
  130 CHOUT=CHOUT-X5
      GO TO 150
  140 CHIN=CHIN+X5
```

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

List of Variables for Module SBCF1F

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,I+1,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+1,I,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 and streams contribute water to the ground-water system or drain water from it depending on the head gradient between the stream and the ground-water regime. The purpose of the River Package is to simulate the effects of flow between surface-water features and ground-water systems. To accomplish this, terms representing seepage to or from the surface features must be added to the ground-water flow equation (equation (26)) for each cell affected by the seepage.

Figure 32 shows a stream divided into reaches so that each reach is completely contained in a single cell. Stream aquifer seepage is simulated between each reach and the model cell that contains that reach.

The cross-section of figure 33-a shows a situation in which the open water of a stream is separated from the ground-water system by a layer of low permeability streambed material. Figure 33-b shows an idealization of this system in which the stream-aquifer interconnection is represented as a simple conductance through which one-dimensional flow occurs. The system of figure 33 is helpful in conceptualizing and describing the simulation of stream-aquifer interaction; however, it must be recognized that, in many instances, no discrete low-permeability streambed layer is present. The techniques of simulation developed through the conceptualization of figure 33 can still be applied to represent these situations, provided the proper interpretation is placed on the various terms and parameters that are used.

Figure 34 shows an isolated view of the idealized streambed conductance of figure 33-b, as it crosses an individual cell. The length of the

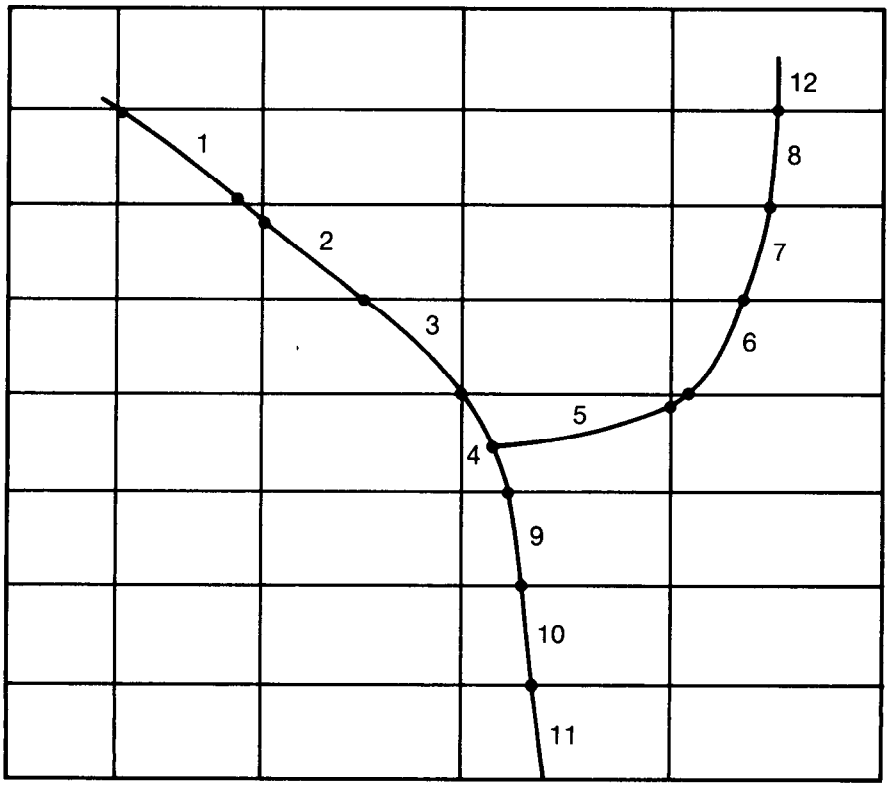
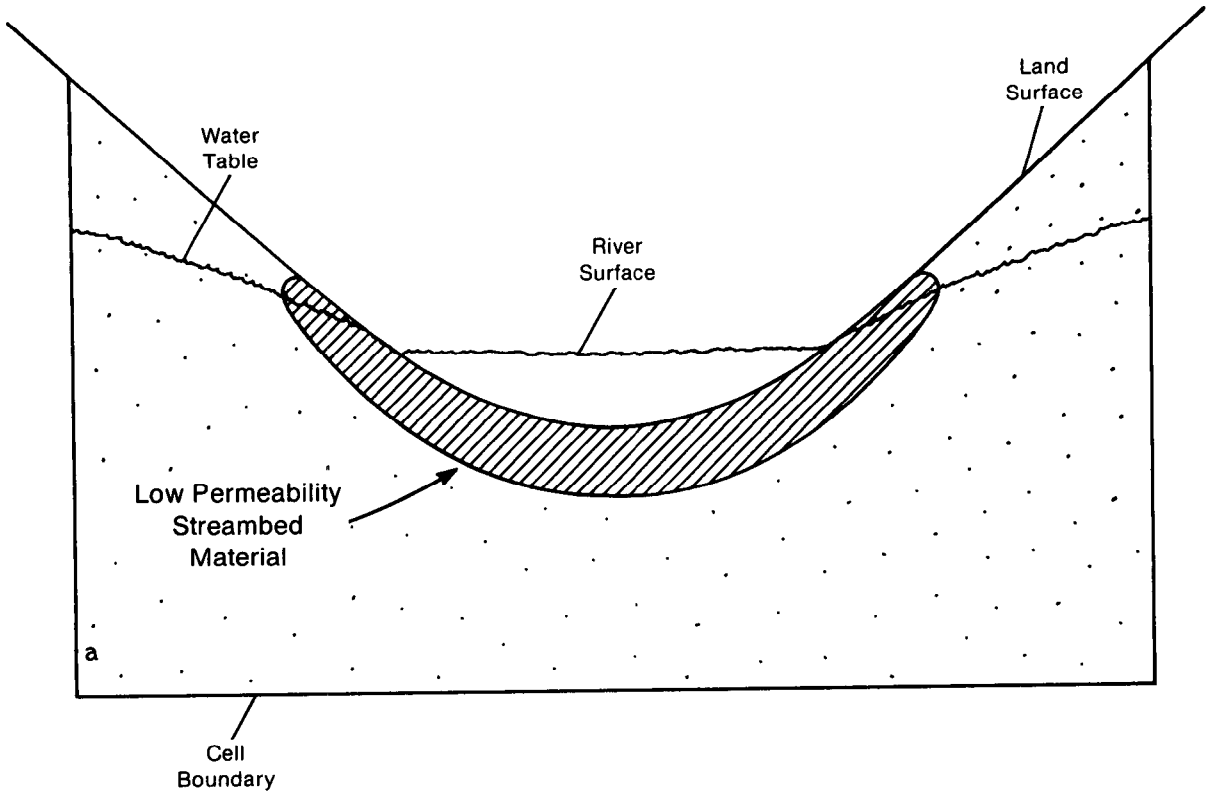


Figure 32.—Discretization of a stream into reaches. Some small reaches are ignored.



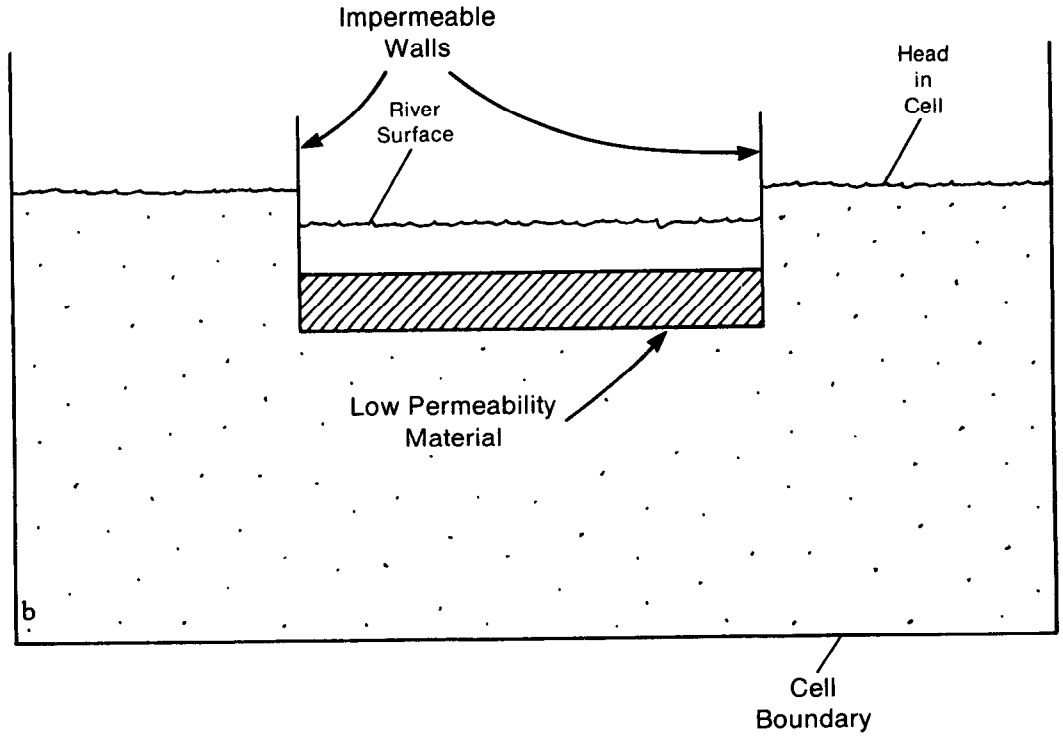


Figure 33.—(a) Cross section of an aquifer containing a stream and (b) Conceptual representation of stream-aquifer interconnection in simulation.

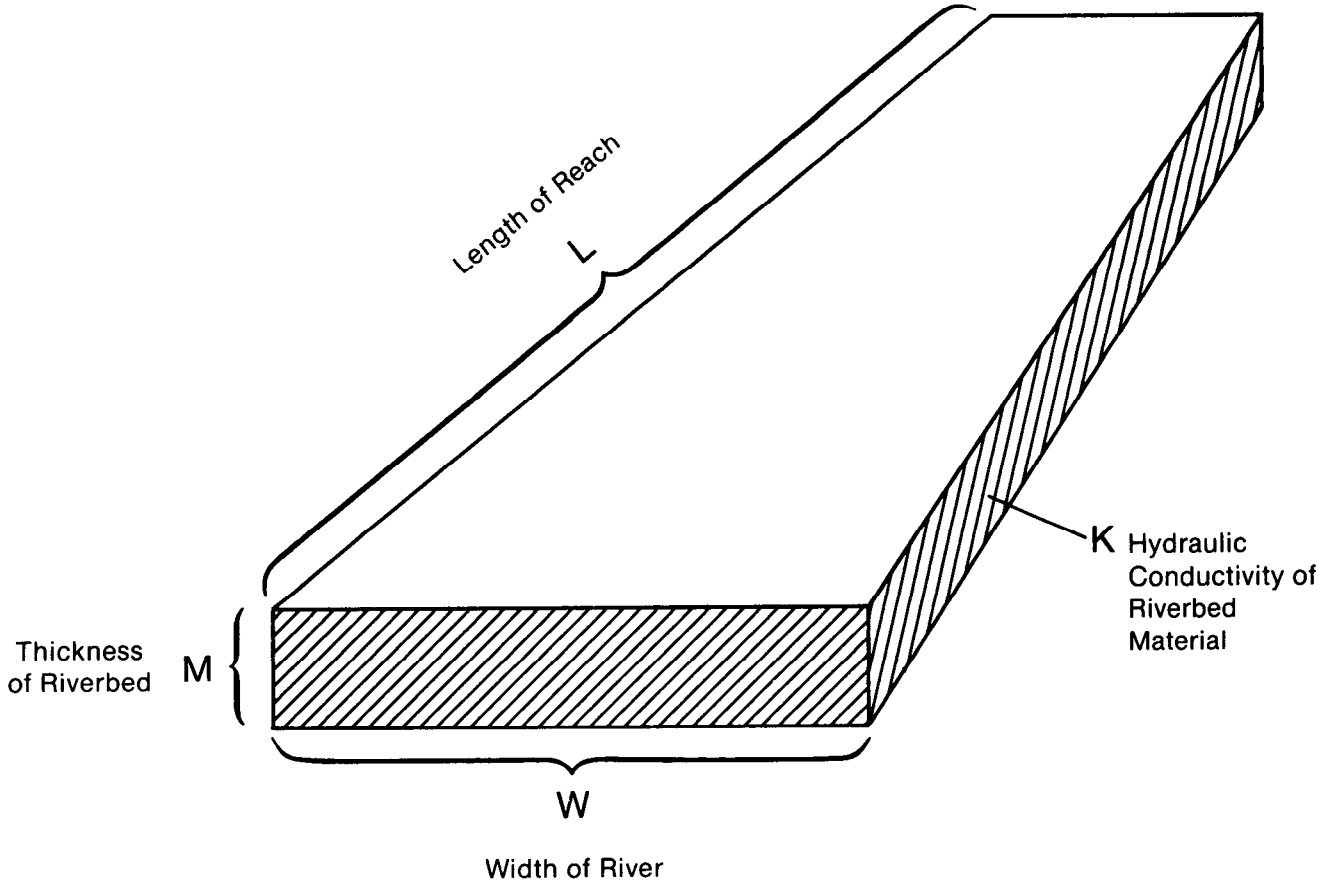


Figure 34.—Idealization of streambed conductance in an individual cell.

conductance block is taken as the length of the stream, L, as it crosses the node; the width is taken as the stream width, W; the distance of flow is taken as the thickness, M, of the streambed layer; and the hydraulic conductivity of the streambed material is designated K. The assumption is made that measurable head losses between the stream and the aquifer are limited to those across the streambed layer itself--that is, that there is no significant head loss between the bottom of the streambed layer and the point represented by the underlying model node. It is further assumed that the underlying model cell remains fully saturated--that is, that its water level does not drop below the bottom of the streambed layer. Under these assumptions, flow between the stream and the ground-water system is given by

$$QRIV = \frac{K \quad L \quad W}{M} \quad (HRIV - h_{i,j,k}) \qquad (63\text{-a})$$

or

$$QRIV = CRIV \ (HRIV - h_{i,j,k}) \qquad (63\text{-b})$$

where QRIV is the flow between the stream and the aquifer, taken as positive if it is directed into the aquifer; HRIV is the head in the stream; CRIV is the hydraulic conductance of the stream-aquifer interconnection (KLW/M), and $h_{i,j,k}$ is the head at the node in the cell underlying the stream reach.

If the assumption is satisfied that all significant head loss occurs across a discrete streambed layer, the application of equations (63) is straightforward. More frequently, however, equations (63) must be applied to situations in which no discrete streambed layer can be identified, or in which head losses are not restricted to those across such a layer. In these cases, the task is to formulate a single conductance term, CRIV, which can be used in (63-b) to relate flow between the stream and the depth

represented by node i,j,k to the corresponding head difference. This flow is in general a three-dimensional process, and its representation through a single conductance term can never be more than approximate. If reliable field measurements of stream seepage and associated head difference are available, they may be used to calculate an effective conductance. Otherwise, a conductance value must be chosen more or less arbitrarily and adjusted during model calibration. Certain rules can be formulated to guide the initial choice of conductance. For example, the assumed cross-sectional area of flow should normally be of the same order of magnitude as the product of channel width and stream reach length within the cell; the assumed distance of flow should not exceed the vertical interval between the streambed and node i,j,k; and, if distinct layers can be recognized within this interval, these should normally be treated as conductances in series in formulating an equivalent conductance. In general, however, it should be recognized that formulation of a single conductance term to account for a three-dimensional flow process is inherently an empirical exercise, and that adjustment during calibration is almost always required.

Equations (63) normally provide an acceptable approximation of stream-aquifer interaction over a certain range of aquifer head values. In most cases, however, if water levels in the aquifer fall below a certain point, seepage from the stream ceases to depend on head in the aquifer. This can be visualized by returning to the concept of a discrete streambed layer. Figure 35-a shows the situation described by equations (63); water level in the aquifer is above the bottom of the streambed layer, and flow through that layer is proportional to the head difference between the stream and the aquifer. In figure 35-b, water level in the aquifer has fallen below the bottom of the streambed layer, leaving an unsaturated interval beneath that

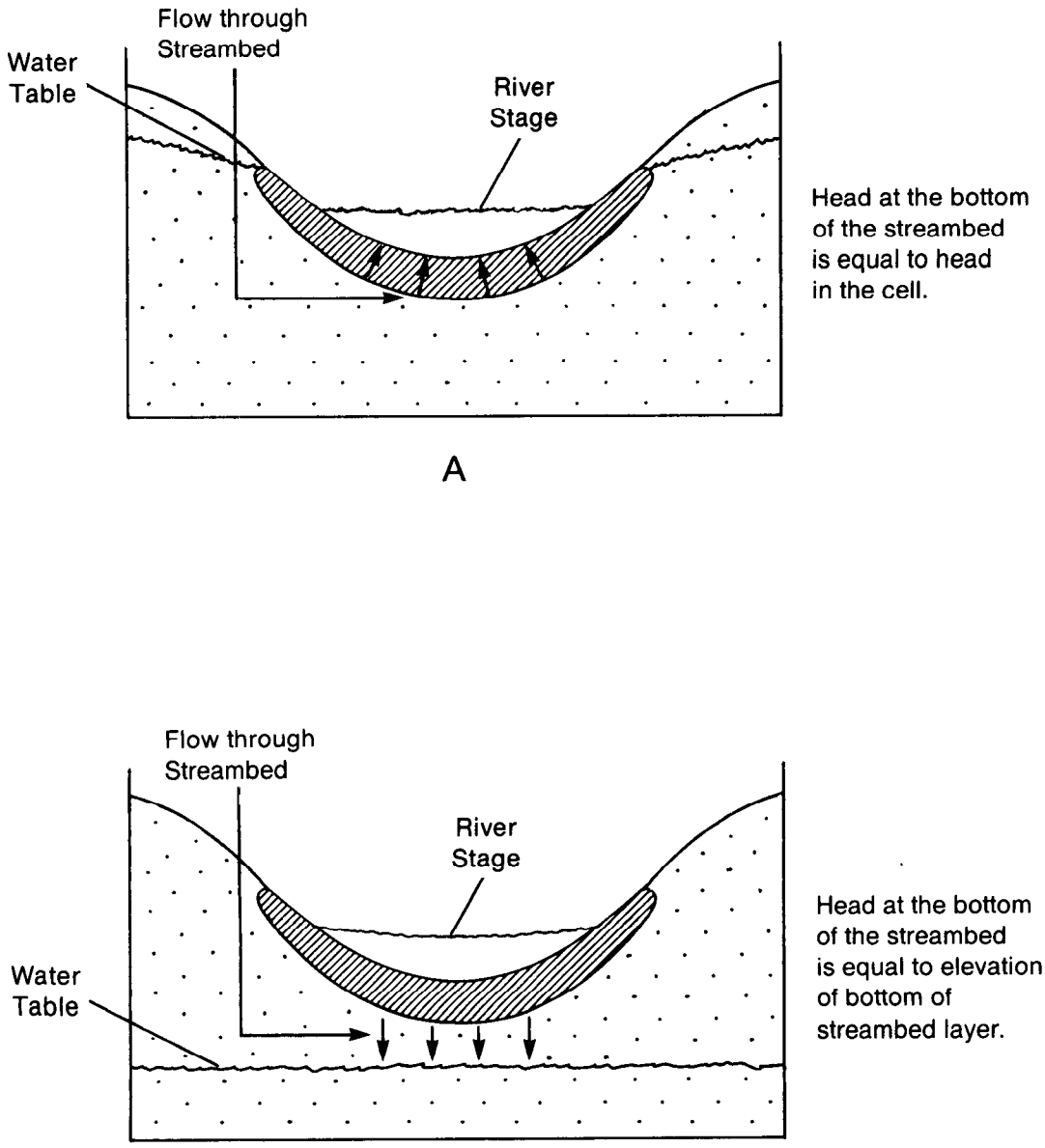


Figure 35.—Cross sections showing the relation between head at the bottom of the streambed layer and head in the cell. Head in the cell is equal to the water-table elevation.

layer; if it is assumed that the streambed layer itself remains saturated, the head at its base will simply be the elevation at that point. If this elevation is designated RBOT, the flow through the streambed layer is given by

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

where QRIV, CRIV, and HRIV are as defined for equation (63-b). Obviously, further declines in head below RBOT produce no increase in flow through the streambed layer; the flow simply retains the constant value given by equation (64), as long as head remains below RBOT. The model described herein utilizes these concepts in simulating stream-aquifer interaction--that is, flow between a stream and a node i,j,k is simulated according to the equation set

$$\begin{aligned} QRIV &= CRIV\ (HRIV - h_{i,j,k}),\ h_{i,j,k} > RBOT \\ QRIV &= CRIV\ (HRIV - RBOT),\ h_{i,j,k} \leq RBOT \end{aligned} \tag{65}$$

Figure 36 shows a graph of flow between the stream and cell i,j,k as a function of the head, $h_{i,j,k}$, as calculated using equations 65. Flow is zero when $h_{i,j,k}$, is equal to the water level in the stream, HRIV. For higher values of $h_{i,j,k}$, flow is negative, that is, into the stream; for lower values of $h_{i,j,k}$, flow is positive, that is, into the aquifer. This positive flow increases linearly as $h_{i,j,k}$ decreases, until $h_{i,j,k}$ reaches RBOT; thereafter the flow remains constant.

A relationship similar to that of equations (65) and figure (36) generally prevails in stream-aquifer interaction whether or not a discrete streambed layer is present. For example, once a break in saturation occurs between the stream and the aquifer, seepage from the stream to the aquifer must become independent of head in the aquifer. In most cases, moreover, this independence is established even before a break in saturation occurs.

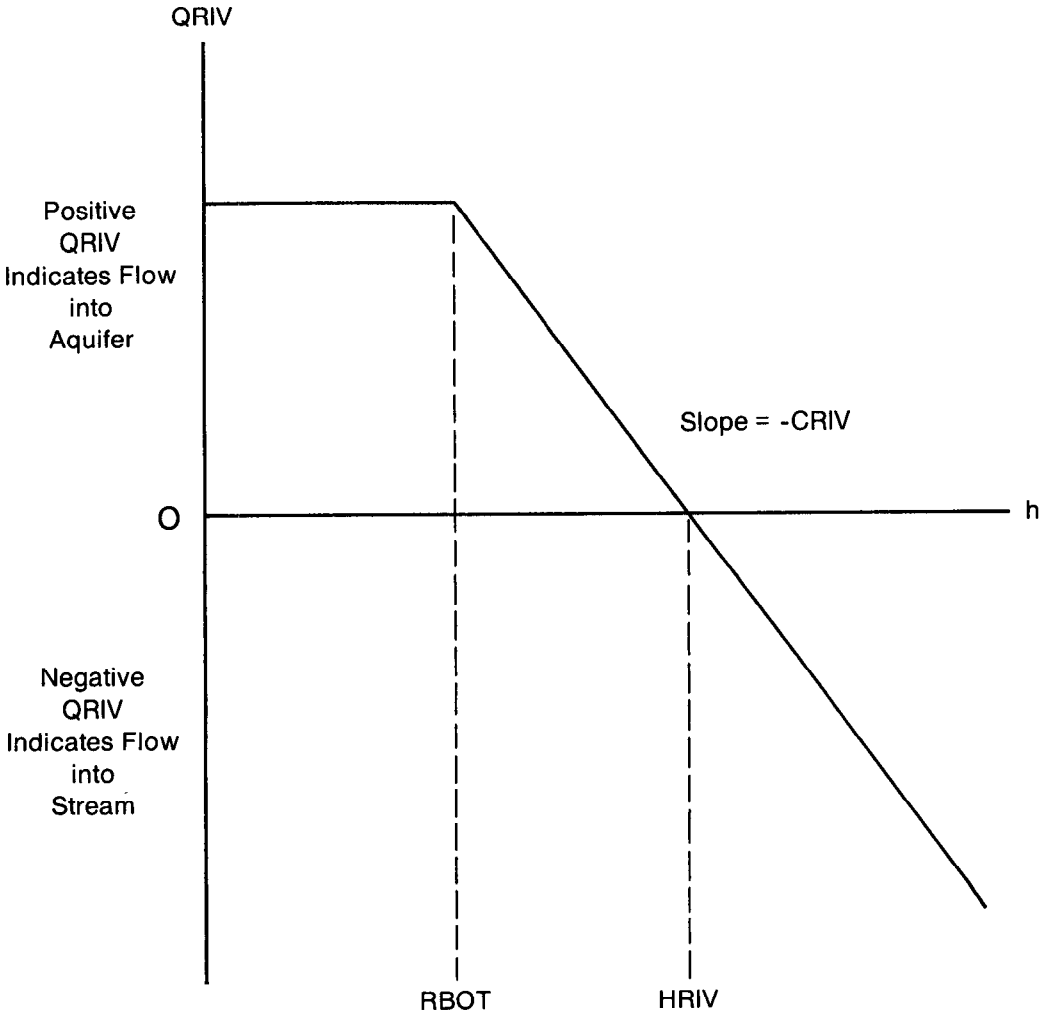


Figure 36.—Plot of flow, QRIV, from a stream into a cell as a function of head, h, in the cell where RBOT is the elevation of the bottom of the streambed and HRIV is the head in the stream.

Figure 37 shows a situation in which levels in the aquifer have fallen far enough below a stream so that only a narrow saturated connection exists between the streambed and the regional water table. Examination of figure 37 will show that the head gradient in the saturated connection must be approximately unity, and that further lowering of the regional water table will not increase this gradient. Thus, once a condition similar to that in figure 37 is established, seepage from the stream is independent of further head decline in the aquifer. The situation shown in figure 37 is itself an overimplification of field conditions, which may often involve complex patters of saturated and unsaturated material beneath the stream. In all situations, however, seepage from the stream must at some point become independent of head in the aquifer, as that head continues to decline.

If hydrologic conditions indicate that seepage from a stream will increase as the local water table elevation declines, but will reach the limiting condition illustrated in figure 37 when the water table reaches an elevation h_1, RBOT should be taken equal to h_1. Because the vertical hydraulic gradient beneath the stream is approximately one under the conditions of figure 37, seepage from the stream into cell i,j,k is given approximately by the product KLW, where K is the vertical hydraulic conductivity of the material in the saturated column beneath the stream, and again L is the length of the stream as it crosses cell i,j,k, and W is stream width. A value of the stream-aquifer conductance term, CRIV, consistent with this estimated value of limiting seepage and with the selected value of RBOT can be obtained by substituting KLW for QRIV in equation (64) and solving for CRIV. This yields

$$CRIV = \frac{KLW}{HRIV-RBOT} \qquad (66)$$

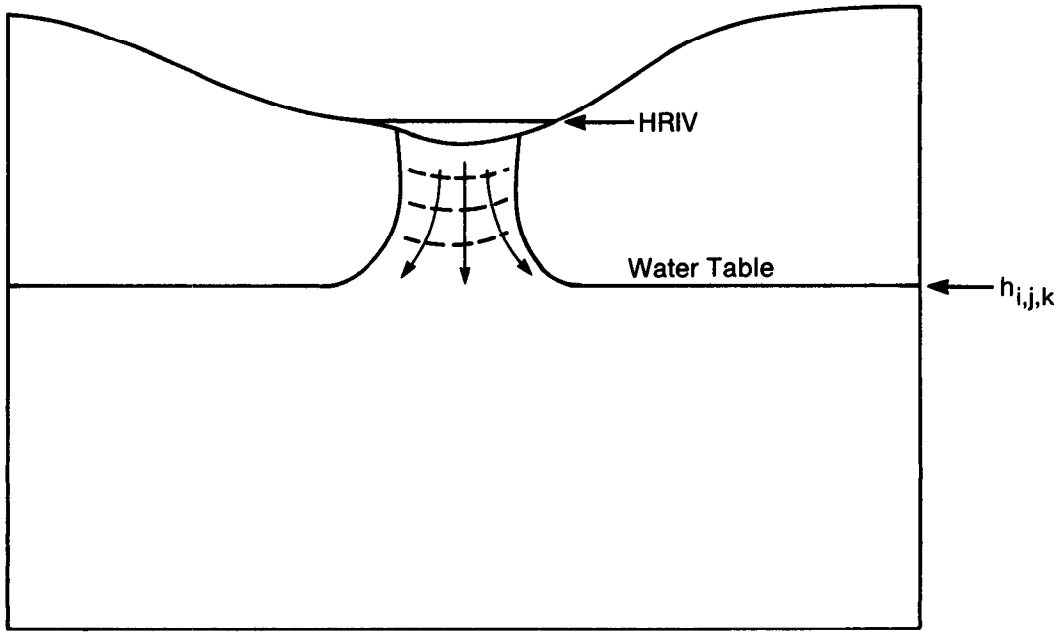


Figure 37.—Limiting seepage from a stream at unit hydraulic gradient

In summary, if the limiting condition of stream seepage is expected to follow the model of figure 37, RBOT should be chosen as the water table elevation at which the transition to this limiting seepage can be expected, and CRIV may be calculated from equation (66). The model simulation technique based on equations (65) should then provide a reasonable approximation to the stream-aquifer interaction.

The simplified model of stream-aquifer interaction utilized here assumes that this interaction is independent of the location of the stream reach within the cell, and that the level of water in the stream is uniform over the reach, and constant over each stress period. The latter assumption implies that conditions of flow in the stream do not vary significantly during the stress period--for example, that the stream does not go dry or overflow its banks, or that such events are of such short duration as to have no effect on stream-aquifer interaction.

Data describing each river are specified by the user for each stress period. Input consists of six entries for each river reach, specifying the layer, row, and column of the cell containing the reach, and the three parameters needed to calculate seepage--stream level or stage (HRIV), the conductance of the stream-aquifer interconnection (CRIV), and the "bottom elevation," or level at which the limiting value of stream seepage is attained (RBOT).

At the start of each iteration, terms representing river seepage are added to the flow equation for each cell containing a river reach. The choice of which river seepage equation to use, equation 63 or equation 64, is made by comparing the most recent value of head at the cell to the value

of RBOT for the reach. Since this process is done at the start of each iteration, the most current value of head (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 63 is selected, the term -CRIV is added to the term HCOF and the term -CRIV*HRIV is added to RHS. If equation 64 is selected, the term -CRIV (HRIV - RBOT) is added to the term RHS.

River Package Input

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

FOR EACH SIMULATION

RIV1AL

1. Data: MXRIVR IRIVCB
   Format: I10 I10

FOR EACH STRESS PERIOD

RIV1RP

2. Data: ITMP
   Format: I10

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

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

Explanation of Fields Used in Input Instructions

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

IRIVCB--is a flag and a unit number.

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

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

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

ITMP--is a flag and a counter.

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

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

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

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

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

Stage--is the head in the river.

Cond--is the riverbed hydraulic conductance.

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

SAMPLE INPUT TO THE RIVER PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS					
1	{MXRIVR, IRIVCB}_____	3	55				
2	{ITMP} FOR FIRST STRESS PERIOD_____	3					
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR FIRST REACH_____	2	6	4	220.	.7	212.
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR SECOND REACH____	2	7	4	225.	.9	217.
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR THIRD REACH_____	2	5	4	210.	.8	200.
2	{ITMP} FOR SECOND STRESS PERIOD_____	-1					
2	{ITMP} FOR THIRD STRESS PERIOD_____	-1					
2	{ITMP} FOR FOURTH STRESS PERIOD_____	2					
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR FIRST REACH_____	2	5	4	210.	.8	200.
3	{LAYER, ROW, COLUMN, STAGE, COND, RBOT} FOR SECOND REACH____	2	6	4	220.	.7	212.
2	{ITMP} FOR FIFTH STRESS PERIOD_____	0					
2	{ITMP} FOR SIXTH STRESS PERIOD_____	-1					

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 and RHS.

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.

Flow Chart for Module RIV1AL

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

MXRIVR is the maximum number of river reaches simulated.

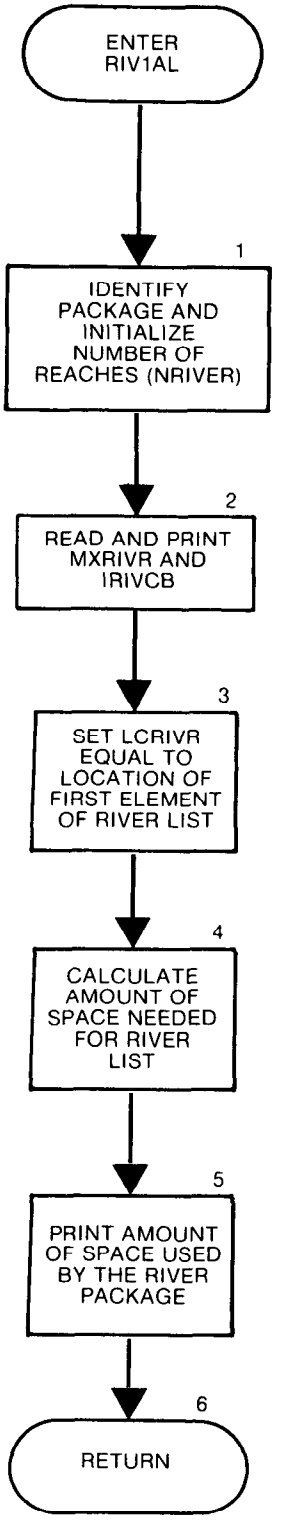
IRIVCB is a flag and a unit number.

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

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

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

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



```
      SUBROUTINE RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,IN,IOUT,
     1                  IRIVCB)
C
C-----VERSION 1554 12MAY1987 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, 9/1/87',
     1' 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,I8,' ELEMENTS IN X ARRAY ARE USED FOR RIVERS')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      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.

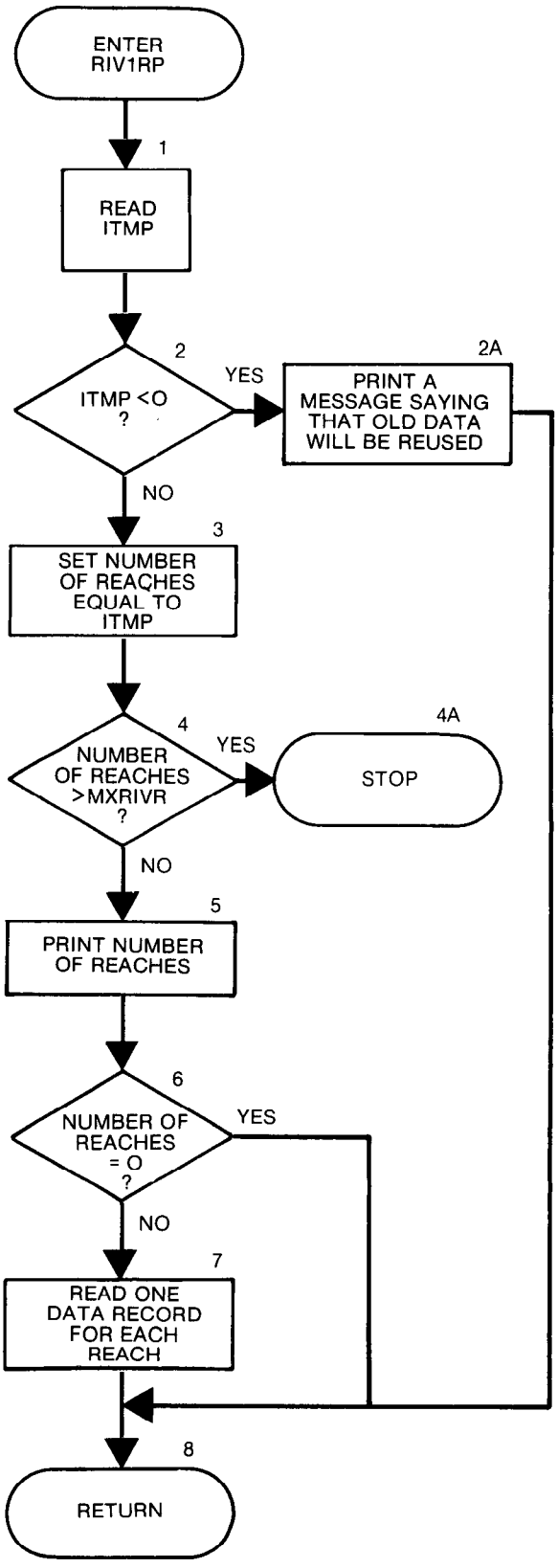
7. Read and print the layer, row, column, riverhead, riverbed conductance, and the elevation of the bottom of the riverbed for each reach.

8. RETURN.

Flow Chart for Module RIV1RP

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

MXRIVR is the maximum number of reaches to be simulated.



```
      SUBROUTINE RIV1RP(RIVR,NRIVER,MXRIVR,IN,IOUT)
C
C
C-----VERSION 1319 25AUG1982 RIV1RP
C     ******************************************************************
C     READ RIVER HEAD, CONDUCTANCE AND BOTTOM ELEVATION
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION RIVR(6,MXRIVR)
C     ------------------------------------------------------------------
C
C
C1------READ ITMP(NUMBER OF RIVER REACHES OR FLAG TO REUSE DATA)
      READ(IN,8)ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP.
      IF(ITMP.GE.0)GO TO 50
C
C2A-----IF ITMP <0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
    7 FORMAT(1HO,'REUSING RIVER REACHES FROM LAST STRESS PERIOD')
      GO TO 260
C
C3------IF ITMP=> ZERO THEN IT IS THE NUMBER OF RIVER REACHES
   50 NRIVER=ITMP
C
C4------IF NRIVER>MXRIVR THEN STOP.
      IF(NRIVER.LE.MXRIVR)GO TO 100
      WRITE(IOUT,99)NRIVER,MXRIVR
   99 FORMAT(1HO,'NRIVER(',I4,') IS GREATER THAN MXRIVR(',I4,')')
C
C4A-----ABNORMAL STOP.
      STOP
C
C5------PRINT NUMBER OF RIVER REACHES IN THIS STRESS PERIOD.
  100 WRITE(IOUT,1)NRIVER
    1 FORMAT(1HO,//1X,I5,' RIVER REACHES')
C
C6------IF THERE ARE NO RIVER REACHES THEN RETURN.
      IF(NRIVER.EQ.0) GO TO 260
C
C7------READ AND PRINT DATA FOR EACH RIVER REACH.
      WRITE(IOUT,3)
    3 FORMAT(1HO,15X,'LAYER',5X,'ROW',5X,'COL   '
     1,'  STAGE     CONDUCTANCE   BOTTOM ELEVATION   RIVER REACH'
     2/1X,15X,80('-'))
      DO 250 II=1,NRIVER
      READ(IN,4)K,I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II)
    4 FORMAT(3I10,3F10.0)
      WRITE(IOUT,5)K,I,J,RIVR(4,II),RIVR(5,II),RIVR(6,II),II
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,G19.4,I10)
      RIVR(1,II)=K
      RIVR(2,II)=I
      RIVR(3,II)=J
  250 CONTINUE
C
C8------RETURN
  260 RETURN
      END
```

List of Variables for Module RIV1RP

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. $\geq$ 0, number of rivers active during the current stress period. < 0, same rivers active during the last stress period will be active during the current stress period.
J	Module	Column number.
K	Module	Layer number.
MXRIVR	Package	Maximum number of river reaches active at any one time.
NRIVER	Package	Number of river reaches active during the current stress period.
RIVR	Package	DIMENSION (6,MXRIVR), For each reach: layer, row, column, river head, riverbed conductance and elevation of bottom of riverbed.

Narrative for Module RIV1FM

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

1. If NRIVER is less than or equal to zero, in the current stress period, there are no river reaches. RETURN.

2. For each reach in the RIVR list, DO STEPS 3-8.

3. Determine the column (IC), row (IR), and layer (IL).

4. If the cell is external (IBOUND(IC, IR, IL) $\leq$ 0), bypass processing on this reach and go on to the next reach.

5. Since the cell is internal, get the river data (riverhead conductance of the riverbed and elevation of the bottom of the riverbed).

6. Compare the head in the aquifer (HNEW) to the elevation of the bottom of the riverbed (RBOT).

7. If the head in the aquifer (HNEW) is greater than the elevation of the bottom of the riverbed (RBOT), add the term -CRIV*HRIV to the accumulator RHS and the term -CRIV to the accumulator HCOF. (CRIV is the riverbed conductance; HRIV is the riverhead.)

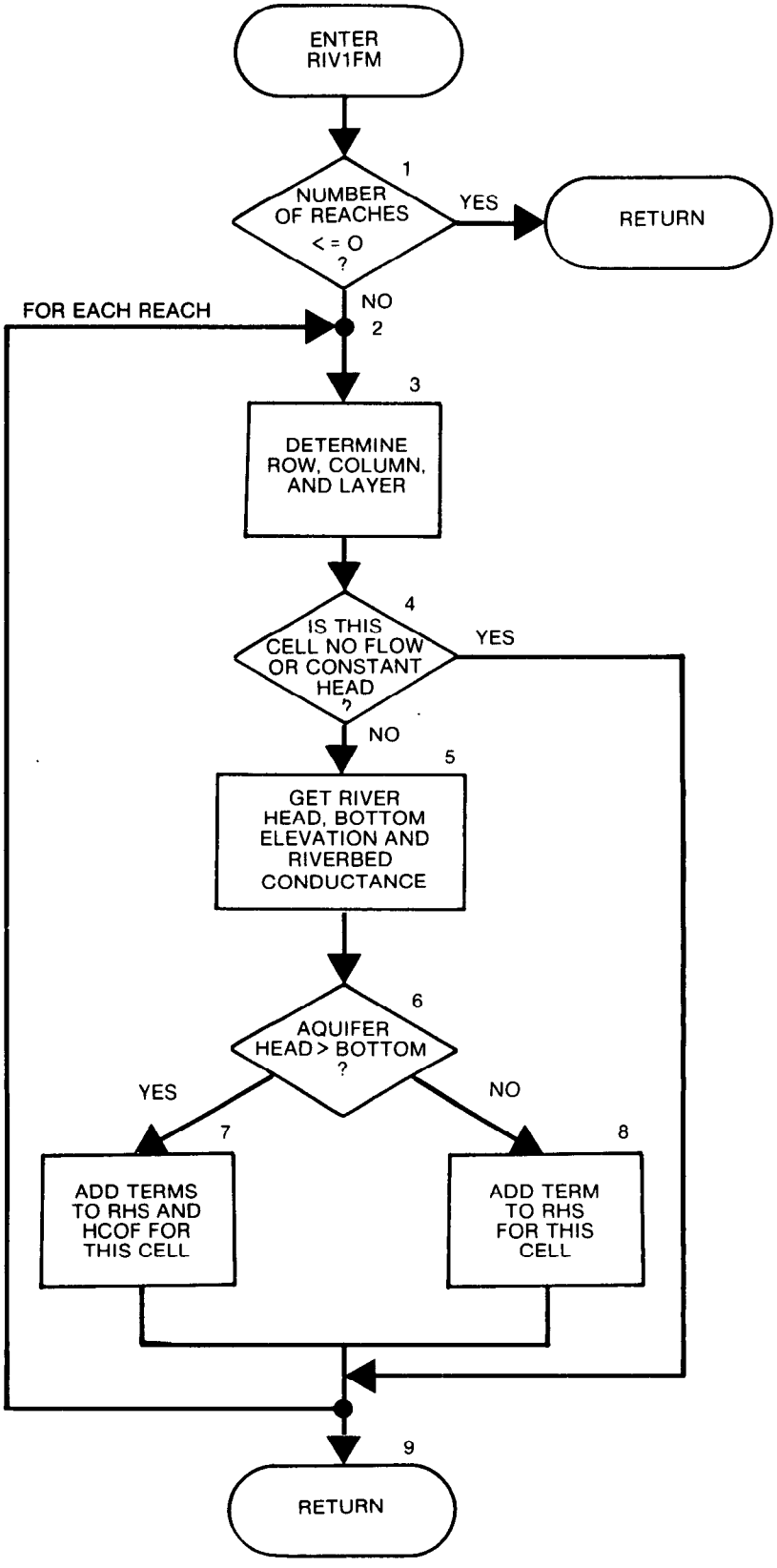
8. If the head in the aquifer (HNEW) is less than or equal to RBOT, add the term -CRIV*(HRIV - RBOT) to the accumulator RHS.

9. RETURN.

Flow Chart for Module RIV1FM

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

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



```
      SUBROUTINE RIV1FM(NRIVER,MXRIVR,RIVR,HNEW,HCOF,RHS,IBOUND,
     1                  NCOL,NROW,NLAY)
C
C-----VERSION 0915 27AUG1982 RIV1FM
C     ******************************************************************
C     ADD RIVER TERMS TO RHS AND HCOF
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      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.

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

17. Move RATIN and RATOUT into the VBVL array for printing by 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.

Flow Chart for Module RIV1BD

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

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

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

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

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

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

IRIVCB is a flag and a unit number.

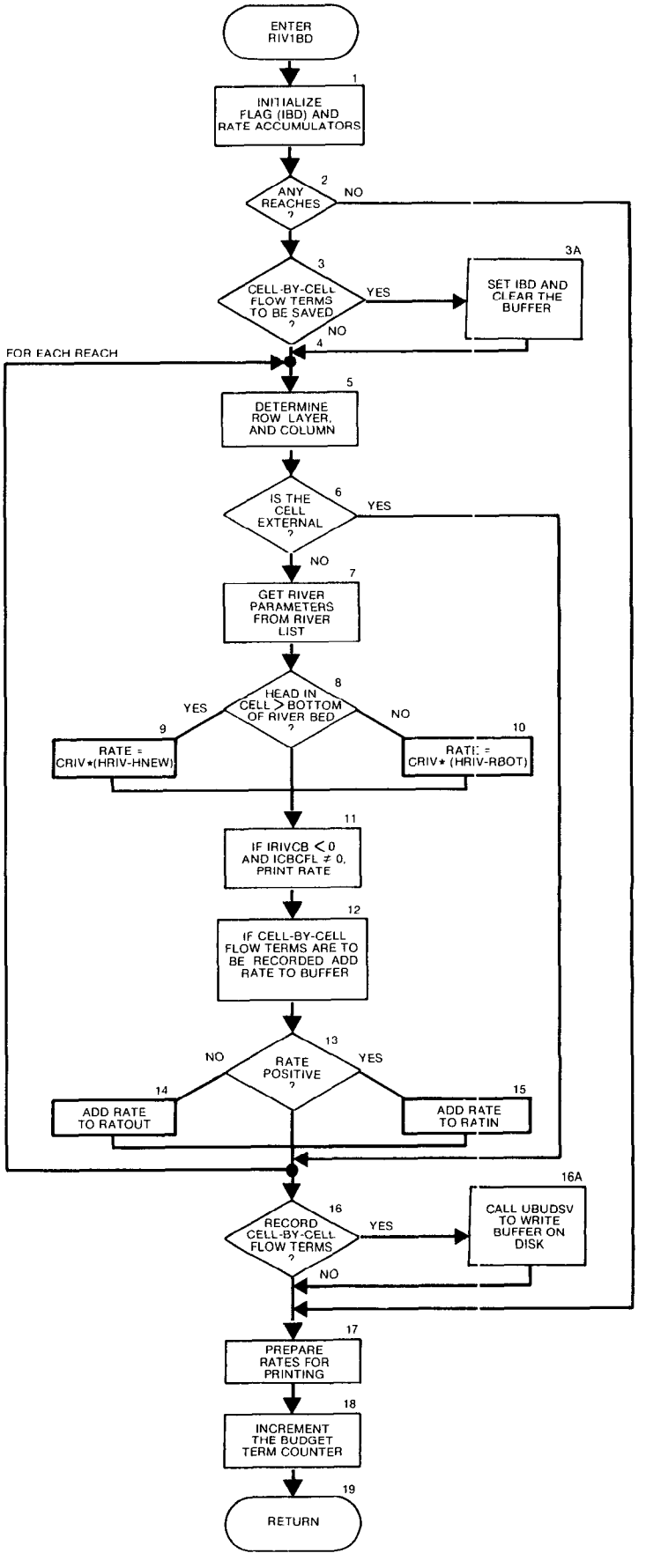
If IRIVCB > 0, it is the unit number on which cell-by-cell flow terms for rivers will be recorded whenever ICBCFL is set.

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

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

ICBCFL is a flag.

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



```
      SUBROUTINE RIV1BD(NRIVER,MXRIVR,RIVR,IBOUND,HNEW,
     1       NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,KSTP,KPER,IRIVCB,
     2       ICBCFL,BUFF,IOUT)
C-----VERSION 1556 12MAY1987 RIV1BD
C     ******************************************************************
C     CALCULATE VOLUMETRIC BUDGET FOR RIVERS
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      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

The Recharge (RCH) Package is designed to simulate areally distributed recharge to the ground-water system. Most commonly, areal recharge occurs as a result of precipitation that percolates to the ground-water system. Recharge applied to the model is defined as

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

where $Q_{Ri,j}$ is the recharge flow rate applied to the model at horizontal cell location (i,j) expressed as a fluid volume per unit time; and $I_{i,j}$ is the recharge flux (in units of length per unit time) applicable to the map area, $DELR_j * DELC_i$, of the cell. The recharge, $Q_{Ri,j}$, is applied to a single cell within the vertical column of cells located at (i j). There is no need to allow for recharge to occur simultaneously at multiple depths in the same vertical column because natural recharge enters the ground-water system at its top. In the simplest situation, the top of the ground-water system will occur in model layer 1; however, the vertical position of the top of the system may vary with horizontal location and with time as the water-table rises and falls. Three options for specifying the cell in each vertical column of cells that receives the recharge have been implemented as described below. The RCH Package can potentially be used to simulate recharge from sources other than precipitation -- for example, artificial recharge. If the ability to apply recharge to more than one cell in a vertical column of cells is required, then the Well Package, which allows recharge or discharge to be specified at any model cell, can be used.

In the package described herein, values of recharge flux, $I_{i,j}$, are read into a two dimensional array, $RECH_{i,j}$, at each stress period (unless an option is exercised to use recharge fluxes from the previous stress period). These values of recharge flux are immediately multiplied by horizontal cell areas, $DELR_j DELC_i$, to obtain values of $Q_{Ri,j}$, which are maintained in the RECH array. The cell within each vertical column to which the recharge is applied is specified through the recharge option code, NRCHOP, and optional array IRCH. The options include: (1) application of the recharge to model layer 1; (2) application of the recharge to any cell in the vertical column as specified by layer numbers contained in two dimensional array $IRCH_{i,j}$; and (3) application of the recharge to the uppermost active cell in the vertical column, provided there is no constant head cell above it in the column. Under options 1 and 2, if a cell designated to receive recharge is no-flow, then no recharge is added. Under the third option, if there is a constant head cell in a vertical column of cells and there is no active cell above, then no recharge is applied to this column because it is assumed that any recharge would be intercepted by the constant head source. Recharge flux values that are read into the model must be expressed in units that are consistent with the length and time units used to represent all other model parameters.

In the formation of the matrix equations, the recharge flow rate, $Q_{Ri,j}$, associated with a given horizontal cell location (i,j) and vertical location, k, that is determind by the recharge option is subtracted from the value of $RHS_{i,j,k}$ (equation (26) or (29)). This is done at each iteration for all cells that receive recharge. Because recharge as defined is independent of aquifer head, nothing is added to the coefficient of head, $HCOF_{i,j,k}$.

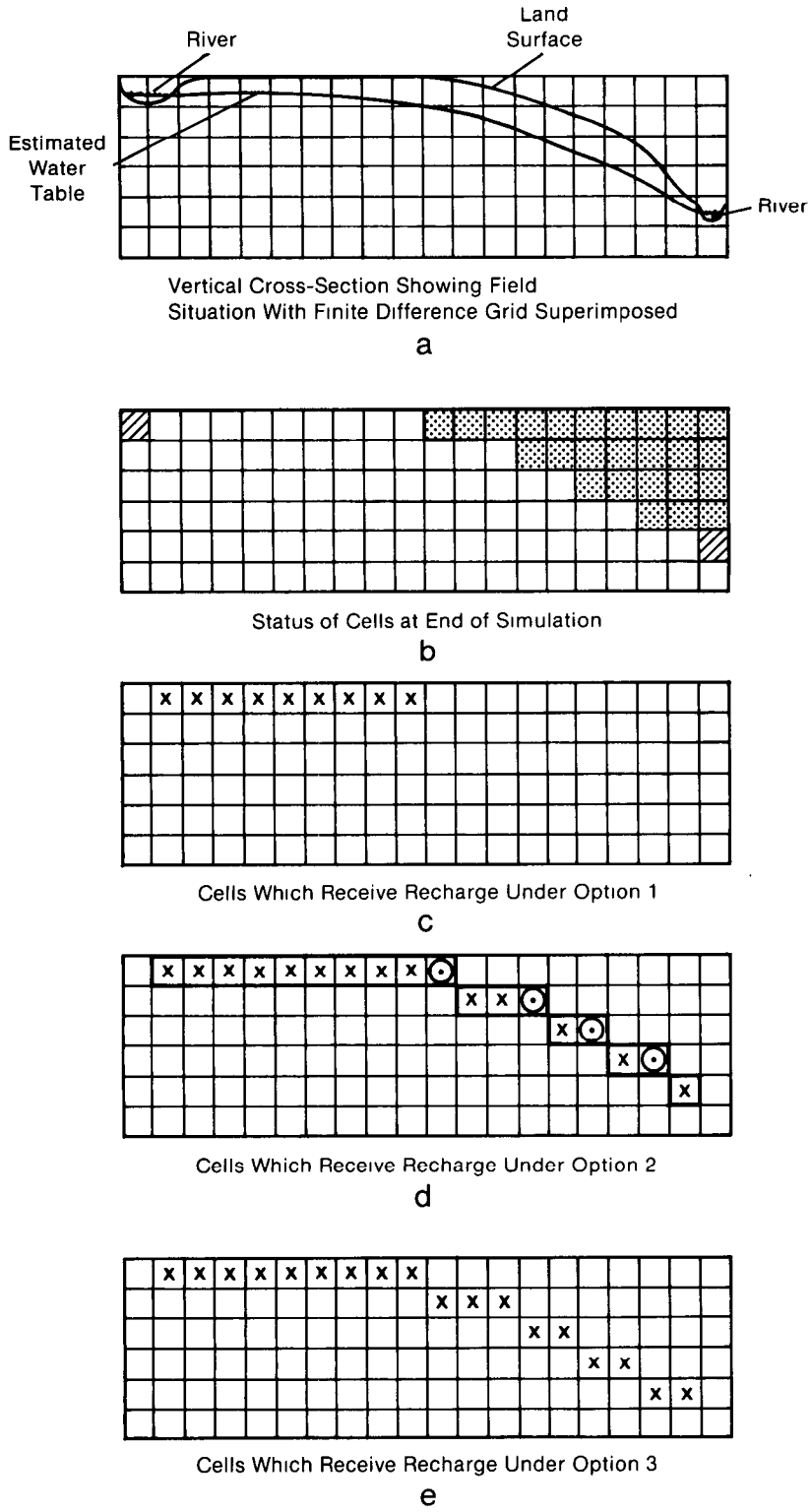


Figure 38.—Hypothetical problem showing which cells receive recharge under the three options available in the Recharge Package.

Careful consideration should be given to the problem under study and to the other options employed in its simulation before deciding which of the three recharge options listed above to utilize in a given situation. For example, figure 38 shows a situation in which a cross sectional model has been used to simulate a hypothetical problem involving recharge, seepage from a stream, and seepage into a stream (figure 38-a). Using the provision described in Chapter 5 for horizontal conductance formulation under water table conditions, the model mesh has been progressively truncated during simulation so that the uppermost active cells in each vertical column fall approximately at the water table. This process yields the final distribution of active, constant head and no-flow cells shown in figure 38-b.

Figure 38-C illustrates the recharge distribution to the model if option 1 above is utilized. Under this option recharge is permitted only to the top layer of the model. Thus once, the water table shape has been simulated by the use of no flow cells in the top layer, recharge to the vertical columns beneath those cells is shut off. This clearly fails to simulate the given system.

Figure 38-d illustrates the recharge distribution if option 2 is utilized, assuming that the user specifies recharge cells prior to the simulation on the basis of an estimated water table position, which differs slightly from that finally obtained in the simulation process. Four of the cells which the user had designated as recharge cells have converted to an inactive condition and thus receive no recharge.

Figure 38-e illustrates the Simulation under the third option, which turns out to be the one best suited for this particular situation. Under this option, recharge enters the uppermost active cell in each vertical column, except where constant head cells have been used to represent the streams. Thus, a continuous distribution of recharge to the water table is simulated.

For the typical situation of recharge from precipitation, option 3 is the easiest to use. The model user does not have to be concerned about determining which is the highest active cell in a vertical column because the program automatically determines this throughout the simulation. Option 1, however, can be useful in situations where recharge should not pass through the no-flow cells in layer 1. For example, some cells may be designated no-flow because they are impermeable. Any recharge specified for those cells should not pass into layer 2. Of course option 3 could still be used in this situation by specifying that the recharge rate is zero at the impermeable cells. The user should select the option that will result in the least effort for specification of input data. Similarly, option 2 may be useful when layers other than layer 1 have outcrop areas and when recharge to the specified layers should not penetrate through no-flow cells to a lower layer. Other factors to consider when choosing the best option are that option 2 uses more computer memory than options 1 and 3, and option 3 uses slightly more computer time than options 1 and 2.

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 fluxes are defined in a two-dimensional array, RECH, with one value for each vertical column. Accordingly, recharge is applied to one cell in each vertical column, and the option code determines which cell in the column is selected for recharge.

1 - Recharge is only to the top grid layer.

2 - Vertical distribution of recharge is specified in array IRCH.

3 - Recharge is applied to the highest active cell in each vertical column. A constant-head node intercepts recharge and prevents deeper infiltration.

IRCHCB--is a flag and a unit number.

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

If IRCHCB $\leq$ 0, cell-by-cell flow terms will not be printed or recorded.

INRECH--is the RECH read flag.

If INRECH $\geq$ 0, an array of recharge fluxes, RECH, is read.

If INRECH < 0, recharge rates from the preceding stress period are used.

INIRCH--is the IRCH read flag. When NRCHOP is two,

If INIRCH $\geq$ 0, an array of layer numbers (IRCH) is read.

If INIRCH < 0, the array (IRCH) used in the preceding stress period is reused.

Note: When NRCHOP is one or three, INIRCH is ignored.

RECH--is the recharge flux (Lt^{-1}). Read only if INRECH is greater than or equal to zero.

IRCH--is the layer number array that defines the layer in each vertical column where recharge is applied. Read only if NRCHOP is two and if INIRCH is greater than or equal to zero.

SAMPLE INPUT TO THE RECHARGE PACKAGE USING RECHARGE OPTION 1

DATA ITEM	EXPLANATION	INPUT RECORDS		
1	{NRCHOP,IRCHCB}	1	0	
2	Stress period 1--{INRECH}	1		
3	Control record for recharge array	12	3.17E-8	(10F4.0) 4
	Recharge rates	1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.1 1.0 1.0 1.0 1.0 1.1 1.0 1.0 1.0 1.1 1.1 1.0 1.0 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1		
2	Stress period 2--{INRECH}	-1		
2	Stress period 3--{INRECH}	-1		
2	Stress period 4--{INRECH}	1		
3	Control record for recharge array	12	3.17E-8	(10F4.0) 4
	Recharge rates	1.2 1.2 1.2 1.2 1.3 1.2 1.2 1.2 1.3 1.4 1.2 1.2 1.3 1.4 1.4 1.0 1.0 1.0 1.1 1.1 1.2 1.3 1.3 1.4 1.4 1.3 1.3 1.4 1.4 1.4		

SAMPLE INPUT TO THE RECHARGE PACKAGE USING RECHARGE OPTION 2

DATA ITEM	EXPLANATION	INPUT RECORDS		
1	{NRCHOP, IRCHCB}	2	0	
2	Stress period 1---{INRECH, INIRCH}	1	1	
3	Control record for recharge array	0	3.17E-8	
4	Control record for layer indicator array	12	1	(20I2) 3
	Layer numbers	1 2 2 2 3 1 2 2 2 2 1 1 2 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1		
2	Stress period 2---{INRECH, INIRCH}	1	-1	
3	Control record for recharge array	0	1.56E-8	
2	Stress period 3---{INRECH, INIRCH}	-1	-1	

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

Module Documentation for the Recharge Package

The Recharge Package (RCH1) consists of four modules, all of which are called by the MAIN program. The modules are:

RCH1AL Allocates space to contain recharge flow rate (RECH) and, if option 2 is specified, the layer-indicator array (IRCH).

RCH1RP Reads recharge flux (in flow per unit area) and indicator array (if option 2 is specified). Multiplies recharge flux by cell area.

RCH1FM Subtract the recharge flow rate from the accumulator in which RHS is formulated.

RCH1BD Calculates the rate and accumulated volume of recharge into the flow system.

Narrative for Module RCH1AL

This module allocates space in the X array to store data relating to areally distributed recharge.

1. Print a message identifying the package.

2. Read and print the option indicator (NRCHOP) and the unit number for cell-by-cell flow terms (IRCHCB).

3. See if the recharge option (NRCHOP) is legal. If NRCHOP is illegal (not 1, 2, or 3), print a message saying the option is illegal. Do not allocate storage. STOP.

4. If NRCHOP is legal, print NRCHOP.

5. If cell-by-cell flow terms are to be recorded, print the unit number where they will be recorded.

6. Allocate space for the recharge array (RECH). Space is allocated by setting the first element of RECH (LCRECH) equal to the location (ISUM) of the first unused element in the X array and adding the size of the array to ISUM.

7. If the recharge option (NRCHOP) is equal to two, allocate space for a layer-indicator array (IRCH).

8. Calculate and print the number of elements in the X array used by the Recharge Package.

9. RETURN.

Flow Chart for Module RCH1AL

NRCHOP is the recharge option.

1 - Recharge is to the top layer.

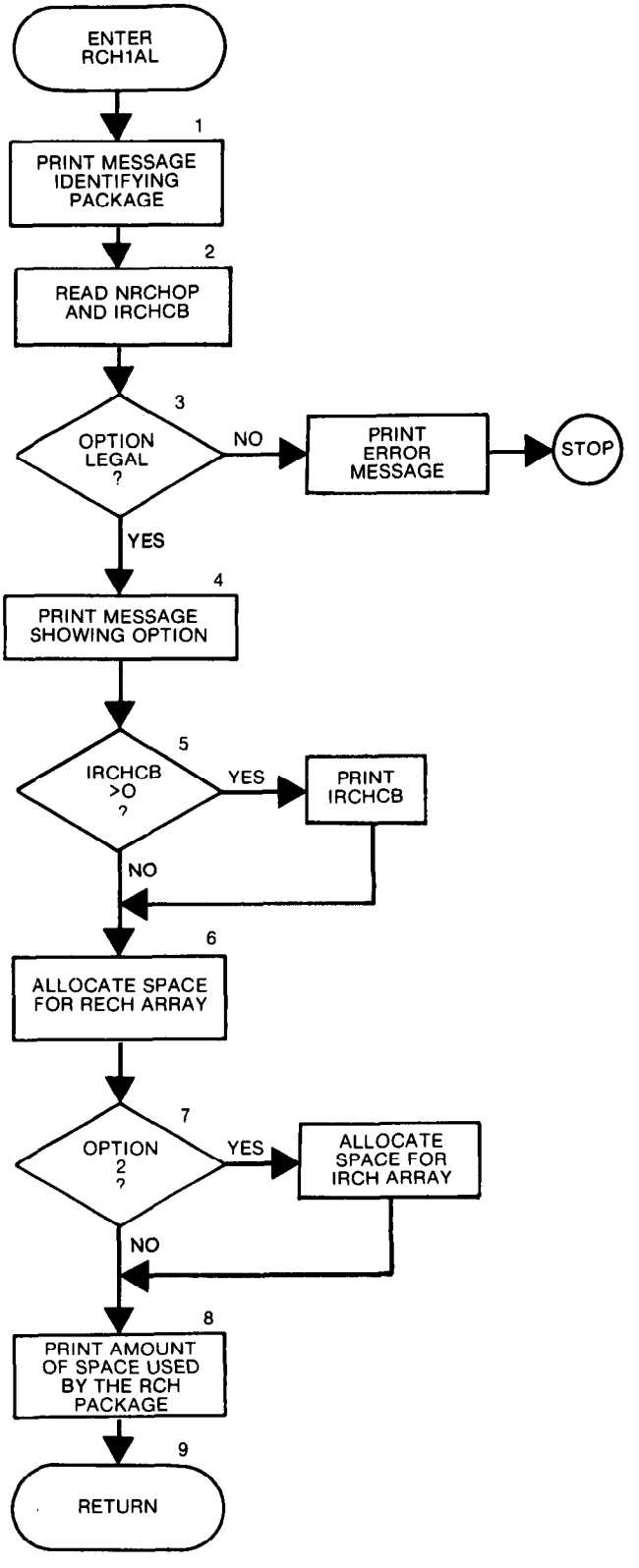
2 - Recharge is to the layer specified by the user in the indicator array (IRCH).

3 - Recharge is to the uppermost active cell.

IRCHCB is the unit number on which cell-by-cell flow terms for recharge will be written.

RECH is an array which contains a recharge rate for each horizontal cell location.

IRCH is an array which contains the layer number to which recharge is applied for each horizontal location. It is used only if option 2 has been specified.



```
      SUBROUTINE RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     1                  NCOL,NROW,IN,IOUT,IRCHCB)
C
C-----VERSION 1559 12MAY1987 RCH1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR RECHARGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE.
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'RCH1 -- RECHARGE PACKAGE, VERSION 1, 9/1/87',
     1' INPUT READ FROM UNIT',I3)
C
C2------READ NRCHOP AND IRCHCB.
      READ(IN,2)NRCHOP,IRCHCB
    2 FORMAT(2I10)
C
C3------CHECK TO SEE THAT OPTION IS LEGAL.
      IF(NRCHOP.GE.1.AND.NRCHOP.LE.3)GO TO 200
C
C3A-----IF ILLEGAL PRINT A MESSAGE AND ABORT SIMULATION
      WRITE(IOUT,8)
    8 FORMAT(1X,'ILLEGAL OPTION CODE. SIMULATION ABORTING')
      STOP
C
C4------IF OPTION IS LEGAL PRINT OPTION CODE.
  200 IRK=ISUM
      IF(NRCHOP.EQ.1) WRITE(IOUT,201)
  201 FORMAT(1X,'OPTION 1 -- RECHARGE TO TOP LAYER')
      IF(NRCHOP.EQ.2) WRITE(IOUT,202)
  202 FORMAT(1X,'OPTION 2 -- RECHARGE TO ONE SPECIFIED NODE IN EACH',
     1        ' VERTICAL COLUMN')
      IF(NRCHOP.EQ.3) WRITE(IOUT,203)
  203 FORMAT(1X,'OPTION 3 -- RECHARGE TO HIGHEST ACTIVE NODE IN EACH',
     1        ' VERTICAL COLUMN')
C
C5------IF CELL-BY-CELL FLOW  TERMS TO BE SAVED THEN PRINT UNIT #
      IF(IRCHCB.GT.0) WRITE(IOUT,204) IRCHCB
  204 FORMAT(1X,'CELL-BY-CELL FLOW TERMS WILL BE RECORDED ON UNIT',I3)
C
C6------ALLOCATE SPACE FOR THE RECHARGE ARRAY(RECH).
      LCRECH=ISUM
      ISUM=ISUM+NCOL*NROW
C
C7------IF OPTION 2 THEN ALLOCATE SPACE FOR INDICATOR ARRAY(IRCH)
      LCIRCH=ISUM
      IF(NRCHOP.NE.2)GO TO 300
      ISUM=ISUM+NCOL*NROW
C
C8------CALCULATE AND PRINT AMOUNT OF SPACE USED BY RECHARGE.
  300 IRK=ISUM-IRK
      WRITE(IOUT,4)IRK
    4 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED FOR RECHARGE')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX)WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE MADE LARGER***')
C
C9------RETURN
      RETURN
      END
```

List of Variables for Module RCH1AL

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 $\leq$ 0, cell-by-cell flow terms will not be printed or recorded. IRCHCB > 0 and ICBCFL $\neq$ 0, cell-by-cell flow terms for the RCH1 Package will be recorded on UNIT = IRCHCB.
IRK	Module	Before this module allocates space, IRK is set equal to ISUM. After allocation, IRK is subtracted from ISUM to get the amount of space in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	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 period are to be used for the current stress period.

2. Test INRECH to see where the recharge flux (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 flux rates by the cell areas to get the volumetric-recharge rate.

5. If the recharge option (NRCHOP) is not equal to two, a layer-indicator array is not needed. GO TO STEP 8.

6. If INIRCH is less than zero, the data in IRCH left over from the last stress period will be used in this stress period. Print a message to that effect. GO TO STEP 8.

7. If INIRCH is greater than or equal to zero, CALL U2DINT to read the IRCH array.

8. RETURN.

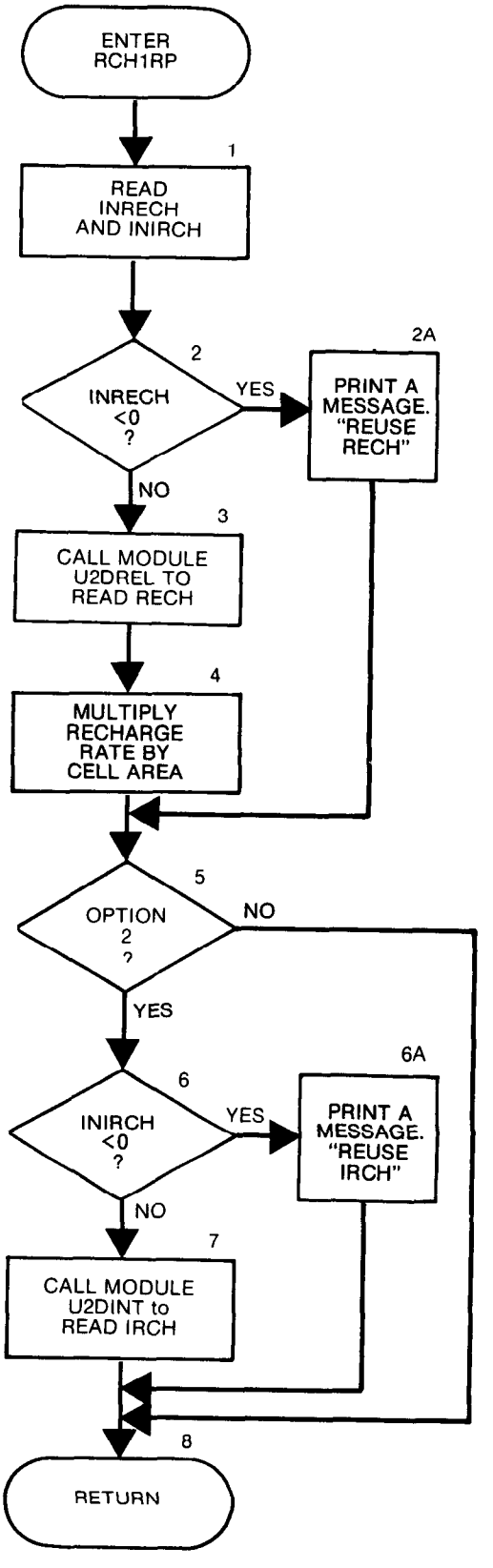
Flow Chart for Module RCH1RP

INRECH is a flag which, when set, indicates that recharge rates (RECH) should be read for the current stress period. If it is clear (< 0), recharge rates from the last stress period should be reused.

INIRCH is a flag similar to INRECH used for the layer indicator array IRCH.

RECH is an array containing a recharge rate for every horizontal cell location.

IRCH is an array containing a recharge indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location which gets recharge. It is used only if the recharge option (NRCHOP) is equal to two.



```
      SUBROUTINE RCH1RP(NRCHOP,IRCH,RECH,DELR,DELC,NROW,NCOL,
     1                  IN,IOUT)
C
C-----VERSION 1634 24JUL1987 RCH1RP
C     ******************************************************************
C     READ RECHARGE RATES
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 ANAME
      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(1HO,'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(1HO,'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.
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 flow rate. Recharge flux is read into RECH and than multiplied by cell area to obtain recharge flow 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 flow 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 flow 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 flow rate from the RHS accumulator. Move on to the next horizontal cell location.

4. RETURN

Flow Chart for Module RCH1FM

RHS is the right hand side of the finite-difference equation. It includes all terms that are independent of head at the end of the time step.

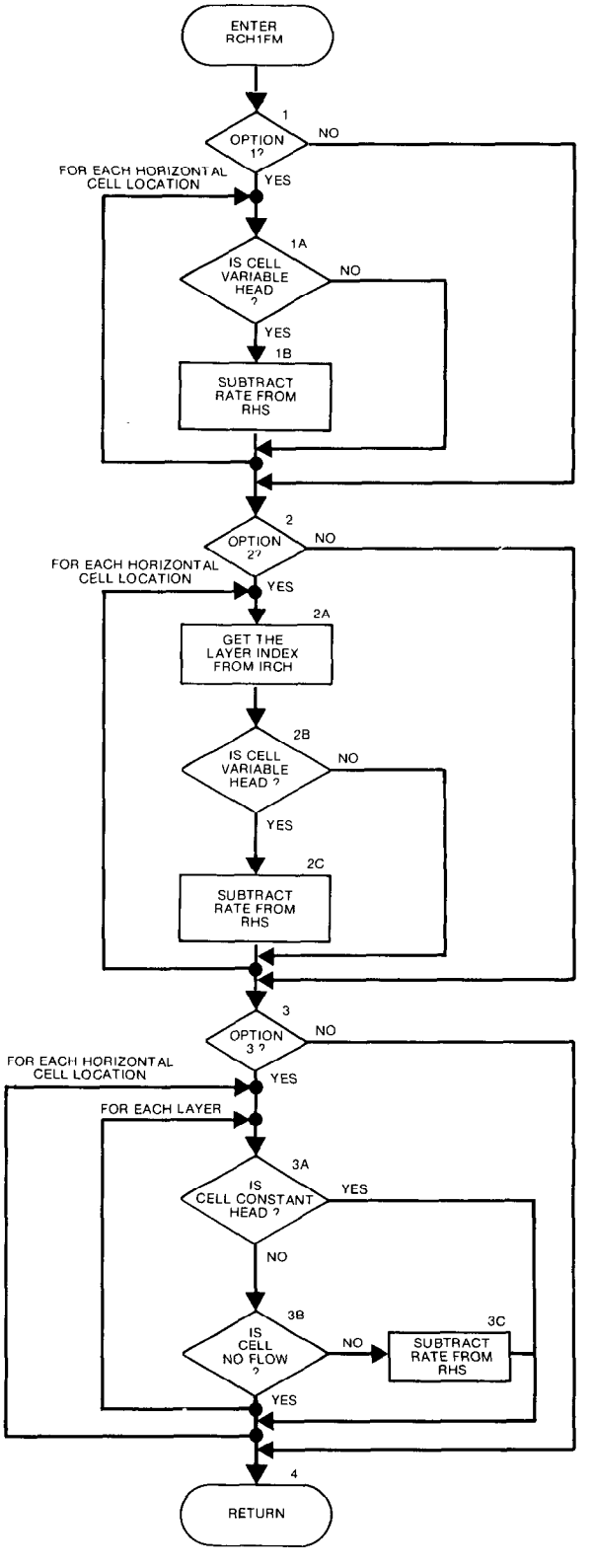
IRCH is an array which contains the layer number to which recharge is applied for each horizontal location. It is used only if option 2 has been specified.

NRCHOP is the recharge option.

1 - Recharge is to the top layer.

2 - Recharge is to the layer specified by the user in the indicator array (IRCH).

3 - Recharge is to the uppermost active cell.



```
      SUBROUTINE RCH1FM(NRCHOP,IRCH,RECH,RHS,IBOUND,NCOL,
     1                  NROW,NLAY)
C
C-----VERSION 1404 12MAY1987 RCH1FM
C     ******************************************************************
C     SUBTRACT RECHARGE FROM RHS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
     1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NRCHOP IS 1 RECHARGE IS IN TOP LAYER. LAYER INDEX IS 1.
      IF(NRCHOP.NE.1) GO TO 15
C
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C1A-----IF CELL IS EXTERNAL THERE IS NO RECHARGE INTO IT.
      IF(IBOUND(IC,IR,1).LE.0)GO TO 10
C
C1B-----SUBTRACT RECHARGE RATE FROM RIGHT-HAND-SIDE.
      RHS(IC,IR,1)=RHS(IC,IR,1)-RECH(IC,IR)
   10 CONTINUE
      GO TO 100
C
C2------IF OPTION IS 2 THEN RECHARGE IS INTO LAYER IN INDICATOR ARRAY
   15 IF(NRCHOP.NE.2)GO TO 25
      DO 20 IR=1,NROW
      DO 20 IC=1,NCOL
C
C2A-----LAYER INDEX IS IN INDICATOR ARRAY.
      IL=IRCH(IC,IR)
C
C2B-----IF THE CELL IS EXTERNAL THERE IS NO RECHARGE INTO IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 20
C
C2C-----SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-RECH(IC,IR)
   20 CONTINUE
      GO TO 100
C
C3------IF OPTION IS 3 RECHARGE IS INTO HIGHEST INTERNAL CELL.
   25 IF(NRCHOP.NE.3)GO TO 100
C        CANNOT PASS THROUGH CONSTANT HEAD NODE
      DO 30 IR=1,NROW
      DO 30 IC=1,NCOL
      DO 28 IL=1,NLAY
C
C3A-----IF CELL IS CONSTANT HEAD MOVE ON TO NEXT HORIZONTAL LOCATION.
      IF(IBOUND(IC,IR,IL).LT.0) GO TO 30
C
C3B-----IF CELL IS INACTIVE MOVE DOWN A LAYER.
      IF (IBOUND(IC,IR,IL).EQ.0)GO TO 28
C
C3C-----SUBTRACT RECHARGE FROM RIGHT-HAND-SIDE.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-RECH(IC,IR)
      GO TO 30
   28 CONTINUE
   30 CONTINUE
  100 CONTINUE
C
C4------RETURN
      RETURN
      END
```

List of Variables for Module RCH1FM

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 flow rate.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

Narrative for Module RCH1BD

This module calculates rates and volumes added to the aquifer by areally distributed recharge.

1. Clear the rate accumulators RATIN and RATOUT.

2. If cell-by-cell flow terms will be saved, clear the buffer (BUFF) in which they will be accumulated.

3. If the recharge option is one, the recharge goes into the top layer. Process the horizontal locations one at a time.

   (a) If the cell is external, do not calculate budget.

   (b) If cell-by-cell flow terms will be saved, add recharge to the buffer.

   (c) If the recharge is positive, add it to RATIN; otherwise, add it to RATOUT.

4. If the recharge option is two, recharge goes into the layer specified in indicator array (IRCH). Process the horizontal locations one at a time.

   (a) Get the cell layer from indicator array (IRCH).

   (b) If the cell is external, do not calculate budget.

   (c) If cell-by-cell flow terms will be saved, add the recharge to the buffer.

   (d) If the recharge is positive, add it to RATIN; otherwise, add it to RATOUT.

5. If the recharge option is three, the recharge goes into the top variable-head cell provided there is not a constant-head cell above it. Process the horizontal locations one at a time. Start with the top cell and work down.

(a) If the cell is inactive, there is no recharge into that cell; move down to the next one.

(b) If the cell is constant, there is no recharge at this horizontal location; move on to the next horizontal location.

(c) If cell-by-cell flow terms are to be saved, add the recharge to the buffer.

(d) If the recharge is positive, add it to RATIN; otherwise, add it to RATOUT.

6. If cell-by-cell flow terms will be saved, call module UBUDSV to write the buffer (BUFF) onto disk.

7. Move RATIN and RATOUT into the VBVL array for printing by BAS1OT.

8. Add RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS1OT.

9. Move the recharge budget-term labels to VBNM for printing by BAS1OT.

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

11. RETURN.

Flow Chart for Module RCH1BD

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

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

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

NRCHOP is the recharge option.

1 - Recharge is to the top layer.

2 - Recharge is to the layer specified by the user in the indicator array (IRCH).

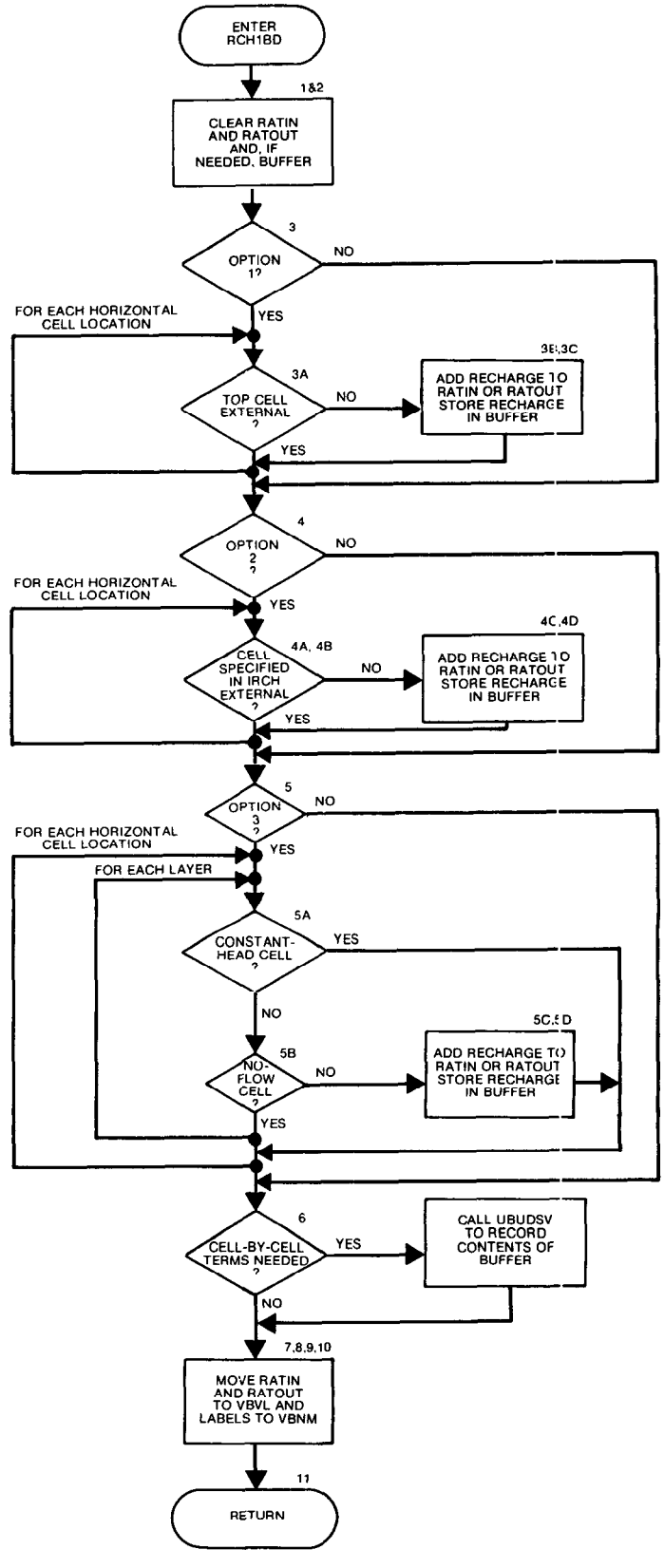
3 - Recharge is to the uppermost active cell.

IRCH is an array containing a recharge indicator for each horizontal cell. It is used only if the recharge option (NRCHOP) is equal to two.

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

VBNM is a table of labels for budget terms.

EXTERNAL: a cell is external if it is either no flow (inactive) or constant head.



```
      SUBROUTINE RCH1BD(NRCHOP,IRCH,RECH,IBOUND,NROW,NCOL,NLAY,
     1    DELT,VBVL,VBNM,MSUM,KSTP,KPER,IRCHCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1602 12MAY1987 RCH1BD
C     ******************************************************************
C     CALCULATE VOLUMETRIC BUDGET FOR RECHARGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      DIMENSION IRCH(NCOL,NROW),RECH(NCOL,NROW),
     1          IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY),
     2          VBVL(4,20),VBNM(4,20)
      DIMENSION TEXT(4)
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','RECH','ARGE'/
C     ------------------------------------------------------------------
C
C1------CLEAR THE RATE ACCUMULATORS.
      RATIN=0.
      RATOUT=0.
C
C2------IF CELL-BY-CELL FLOW TERMS WILL BE SAVED THEN CLEAR THE BUFFER.
      IBD=0
      IF(ICBCFL.EQ.0 .OR. IRCHCB.LE.0) GO TO 5
      IBD=1
      DO 2 IL=1,NLAY
      DO 2 IR=1,NROW
      DO 2 IC=1,NCOL
      BUFF(IC,IR,IL)=0.
    2 CONTINUE
C
C3------IF NRCHOP=1 RECH GOES INTO LAYER 1. PROCESS EACH HORIZONTAL
C3------CELL LOCATION.
    5 IF(NRCHOP.NE.1) GO TO 15
C
C     ---RECHARGE IS APPLIED TO TOP LAYER
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C3A-----IF CELL IS EXTERNAL THEN DO NOT DO BUDGET FOR IT.
      IF(IBOUND(IC,IR,1).LE.0)GO TO 10
      Q=RECH(IC,IR)
C
C3B-----IF CELL-BY-CELL FLOW TERMS WILL BE SAVED THEN ADD RECH TO BUFF
      IF(IBD.EQ.1) BUFF(IC,IR,1)=Q
C
C3C-----IF RECH POSITIVE ADD IT TO RATIN ELSE ADD IT TO RATOUT.
      IF(Q) 8,10,7
    7 RATIN=RATIN+Q
      GO TO 10
    8 RATOUT=RATOUT-Q
   10 CONTINUE
      GO TO 100
C
C4------IF NRCHOP=2 RECH IS IN LAYER SHOWN IN INDICATOR ARRAY(IRCH).
C4------PROCESS HORIZONTAL CELL LOCATIONS ONE AT A TIME.
   15 IF(NRCHOP.NE.2)GO TO 25
      DO 20 IR=1,NROW
      DO 20 IC=1,NCOL
C
C4A-----GET LAYER INDEX FROM INDICATOR ARRAY(IRCH).
      IL=IRCH(IC,IR)
C
C4B-----IF CELL IS EXTERNAL DO NOT CALCULATE BUDGET FOR IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 20
```

```
      Q=RECH(IC,IR)
C
C4C-----IF C-B-C FLOW TERMS WILL BE SAVED THEN ADD RECHARGE TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=Q
C
C4D-----IF RECHARGE IS POSITIVE ADD TO RATIN ELSE ADD IT TO RATOUT.
      IF(Q) 18,20,17
   17 RATIN=RATIN+Q
      GO TO 20
   18 RATOUT=RATOUT-Q
   20 CONTINUE
      GO TO 100
C
C5------IF OPTION=3 RECHARGE IS INTO HIGHEST INTERNAL CELL. IT WILL NOT
C5------PASS THROUGH A CONSTANT HEAD CELL. PROCESS HORIZONTAL CELL
C5------LOCATIONS ONE AT A TIME.
   25 IF(NRCHOP.NE.3)GO TO 100
      DO 30 IR=1,NROW
      DO 30 IC=1,NCOL
      DO 28 IL=1,NLAY
C
C5A-----IF CELL IS CONSTANT HEAD MOVE ON TO NEXT HORIZONTAL LOCATION.
      IF(IBOUND(IC,IR,IL).LT.0) GO TO 30
C
C5B-----IF CELL IS INACTIVE MOVE DOWN TO NEXT CELL.
      IF (IBOUND(IC,IR,IL).EQ.0)GO TO 28
      Q=RECH(IC,IR)
C
C5C-----IF C-B-C FLOW TERMS TO BE SAVED THEN ADD RECHARGE TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=Q
C
C5D-----IF RECH IS POSITIVE ADD IT TO RATIN ELSE ADD IT TO RATOUT.
      IF(Q) 27,30,26
   26 RATIN=RATIN+Q
      GO TO 30
   27 RATOUT=RATOUT-Q
      GO TO 30
   28 CONTINUE
   30 CONTINUE
C
  100 CONTINUE
C
C6------IF C-B-C FLOW TERMS TO BE SAVED CALL UBUDSV TO WRITE THEM.
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IRCHCB,BUFF,NCOL,NROW,
     1                         NLAY,IOUT)
C
C7------MOVE TOTAL RECHARGE RATE INTO VBVL FOR PRINTING BY BAS1OT.
      VBVL(4,MSUM)=RATOUT
      VBVL(3,MSUM)=RATIN
C
C8------ADD RECHARGE FOR TIME STEP TO RECHARGE ACCUMULATOR IN VBVL.
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
C
C9------MOVE BUDGET TERM LABELS TO VBNM FOR PRINT BY MODULE BAS_OT.
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C10-----INCREMENT BUDGET TERM COUNTER.
      MSUM=MSUM+1
C
C11-----RETURN
      RETURN
      END
```

List of Variables for Module RCH1BD

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.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be recorded for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IRCH	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal cell location to which recharge will be applied if the recharge option (NRCHOP) is equal to 2.
IRCHCB	Package	Flag. IRCHCB ≤ 0, cell-by-cell flow terms will not be recorded or printed. IRCHCB > 0 and ICBCFL ≠ 0, cell-by-cell flow terms for the RCH1 Package will be recorded on UNIT = IRCHCB.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.

List of Variables for Module RCH1BD (Continued)

Variable	Range	Definition
NLAY	Global	Number of layers in the grid.
NRCHOP	Package	Recharge option: = 1, recharge is to the top grid layer. = 2, recharge is to the grid layer specified in array IRCH. = 3, recharge is to the highest variable-head cell which is not below a constant-head cell.
NROW	Global	Number of rows in the grid.
Q	Module	Flow from recharge into a cell. (Reverse the sign to get flow out of the cell.)
RATIN	Module	Accumulator for the total flow into the flow field from recharge.
RATOUT	Module	Accumulator for the total flow out of the flow field to recharge.
RECH	Package	DIMENSION (NCOL,NROW), Recharge flow rate.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the current time step out of the flow field. (3,N), Volume into the flow field during simulation. (4,N), Volume out of the flow field during simulation.

CHAPTER 8

WELL PACKAGE

Conceptualization and Implementation

The Well Package is designed to simulate features such as wells which withdraw water from the aquifer (or add water to it) at a specified rate during a given stress period, where the rate is independent of both the cell area and the head in the cell. The discussion in this section is developed on the assumption that the features to be simulated are actually wells, either discharging or recharging.

Well discharge is handled in the Well Package by specifying the rate, Q, at which each individual well adds water to the aquifer or removes water from it, during each stress period of the simulation. Negative values of Q are used to indicate well discharge, while positive values of Q indicate a recharging well.

At the beginning of each stress period, the WEL1RP module reads four values for each well--the row, column and layer number of the cell in which the well is located, and the discharge or recharge rate, Q, of the well during that stress period. At each iteration, as the matrix equations are formulated, the value of Q for each well is subtracted from the RHS value (equation (26) or (29)) for the cell containing that well. Where more than one well falls within a single cell, the calculation is repeated for each well as the RHS term for that cell is assembled. Thus the user specifies the discharge associated with each individual well, and these are in effect summed within the program to obtain the total discharge from the cell.

The Well Package, as it is presently formulated, does not accommodate wells which are open to more than one layer of the model. However, a well of this type can be represented as a group of single-layer wells, each open to one of the layers tapped by the multi-layer well, and each having an individual Q term specified for each stress period. If this approach is used, the discharge of the multi-layer well must be divided or apportioned in some way among the individual layers, externally to the model program. A common method of doing this is to divide the well discharge in proportion to the layer transmissivities i.e.

$$\frac{Q_l}{Q_w} = \frac{T_l}{\Sigma T} \tag{68}$$

where Q_l is the discharge from layer l to a particular well in a given stress period, Q_w is the well discharge in that stress period, T_l is the transmissivity of layer l and ΣT represents the sum of the transmissivities of all layers penetrated by the well. Again, it's important to note that equation (68), or some other method of apportioning the discharge, must be implemented by the user externally to the program for each multi-layer well, and for each stress period.

This approach, in which a multi-layer well is represented as a group of single layer wells, fails to take into account the interconnection between various layers provided by the well itself, and is thus an incomplete representation of the problem. A package which will provide an improved approximation of multi-layer well effects is under development.

Well Package Input

Input for the Well (WEL) Package is read from the unit specified in IUNIT(2).

FOR EACH SIMULATION

WEL1AL

1. Data: MXWELL IWELCB
   Format: I10 I10

FOR EACH STRESS PERIOD

WEL1RP

2. Data: ITMP
   Format: I10

3. Data: Layer Row Column Q
   Format: I10 I10 I10 F10.0

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

Explanation of Fields Used in Input Instructions

MXWELL--is the maximum number of wells used at any time.

IWELCB--is a flag and a unit number.

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

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

If IWELCB < 0, well recharge will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

Layer--is the layer number of the model cell that contains the well.

Row--is the row number of the model cell that contains the well.

Column--is the column number of the model cell that contains the well.

Q--is the volumetric recharge rate. A positive value indicates recharge and a negative value indicates discharge.

SAMPLE INPUT TO THE WELL PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS			
1	{MXWELL, IWELCB}	6	24		
2	STRESS PERIOD 1 {ITMP}	4			
3	FIRST WELL {Layer, Row, Column, Q	2	5	6	-.17
3	SECOND WELL {Layer, Row, Column, Q}	2	6	3	-.23
3	THIRD WELL {Layer, Row, Column, Q}	2	7	7	-.77
3	FOURTH WELL {Layer, Row, Column, Q}	2	9	2	-.32
2	STRESS PERIOD 2 {ITMP}	-1			
2	STRESS PERIOD 3 {ITMP}	-1			
2	STRESS PERIOD 4 {ITMP}	6			
3	FIRST WELL {Layer, Row, Column, Q}	2	6	5	-.73
3	SECOND WELL {Layer, Row, Column, Q}	2	9	4	-.32
3	THIRD WELL {Layer, Row, Column, Q	2	5	6	-.17
3	FOURTH WELL {Layer, Row, Column, Q}	2	6	3	-.43
3	FIFTH WELL {Layer, Row, Column, Q}	2	7	7	-.77
3	SIXTH WELL {Layer, Row, Column, Q}	2	9	2	-.32

Module Documentation for the Well Package

The Well Package (WEL1) consists of four modules, all of which are called by the MAIN program. The modules are:

WEL1AL — Allocates space for the list of wells (WELL).

WEL1RP — Reads location and Q value (discharge or recharge rate) for all wells.

Note: Q is entered as a negative number for well discharge and as a positive number for well recharge.

WEL1FM — Subtracts Q values from the term RHS for each cell containing pumping wells.

WEL1BD — Calculates the rates and accumulated volume of recharge to or discharge from the flow system by pumping wells.

Narrative for Module WEL1AL

This module allocates space in the X array to store the list of wells. The X array is a pool of memory space from which space is allocated for tables, lists, and arrays.

1. Print a message identifying the package and initialize NWELLS (a counter containing the number of wells).

2. Read and print MXWELL (the maximum number of wells) and IWELBD (the unit number for cell-by-cell flow terms or a flag indicating that cell-by-cell flow terms should be printed).

3. Set LCWELL, which will point to the first element in the well list (WELL), equal to ISUM, which is currently pointing to the first unallocated element in the X array.

4. Calculate the amount of space needed for the well list (four values for each cell--row, column, layer, and rate) and add it to ISUM.

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

6. If the pointer to the lowest unallocated element in the X array (ISUM) is greater than the length of the X array (LENX), print a message warning that the X array will have to be enlarged.

7. RETURN.

Flow Chart for Module WEL1AL

MXWELL is the maximum number of wells that will be active at any one time during the simulation.

IWELCB is a flag and a unit number.

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

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

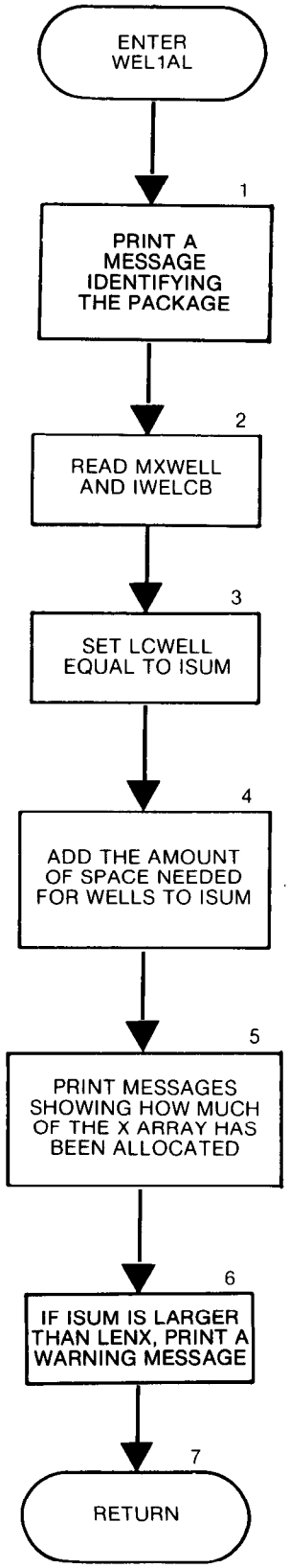
If IWELCB < 0, well recharge will be printed whenever ICBCFL is set.

LCWELL is a location pointer to the first storage location occupied by the well list.

ISUM is the location of the lowest unallocated storage location in the X array.

X array is the pool of memory space allocated for storing specific tables, arrays, and lists.

LENX is the size of the X array.



```
      SUBROUTINE WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,IN,IOUT,
     1                  IWELCB)
C
C-----VERSION 1538 12MAY1987 WEL1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR WELL PACKAGE
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE NWELLS
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'WEL1 -- WELL PACKAGE, VERSION 1, 9/1/87',
     1' INPUT READ FROM',I3)
      NWELLS=0
C
C2------READ MAX NUMBER OF WELLS AND
C2------UNIT OR FLAG FOR CELL-BY-CELL FLOW TERMS.
      READ(IN,2) MXWELL,IWELCB
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXWELL
    3 FORMAT(1H ,'MAXIMUM OF',I5,' WELLS')
      IF(IWELCB.GT.0) WRITE(IOUT,9) IWELCB
    9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IWELCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3------SET LCWELL EQUAL TO LOCATION OF WELL LIST IN X ARRAY.
      LCWELL=ISUM
C
C4------ADD AMOUNT OF SPACE USED BY WELL LIST TO ISUM.
      ISP=4*MXWELL
      ISUM=ISUM+ISP
C
C5------PRINT NUMBER OF SPACES IN X ARRAY USED BY WELL PACKAGE.
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED FOR WELLS')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
C
C6------IF THERE ISN'T ENOUGH SPACE IN THE X ARRAY THEN PRINT
C6------A WARNING MESSAGE.
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C7------RETURN
      RETURN
      END
```

List of Variables for Module WEL1AL

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.
ISP	Module	Number of words in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM-1.
IWELCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, well recharge will be printed whenever ICBCFL is set.
LCWELL	Package	Location in the X array of the first element of array WELL.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXWELL	Package	Maximum number of wells active at any one time.
NWELLS	Package	Number of wells active during the current stress period.

Narrative for Module WEL1RP

This module reads data to build the WELL list.

1. Read ITMP.

(a) If ITMP is less than zero, the well data read for the last stress period will be reused. Print a message to that effect and RETURN.

(b) If ITMP is greater than or equal to zero, it is equal to the number of wells (NWELLS) in the current stress period.

2. If the number of wells (NWELLS) in the current stress period is greater than the number specified as the maximum for the simulation (MXWELL), STOP.

3. Print the number of wells in the current stress period (NWELLS).

4. If there are no wells in the current stress period (NWELLS), bypass further well processing.

5. For each well, read and print the layer, row, column, and well recharge rate.

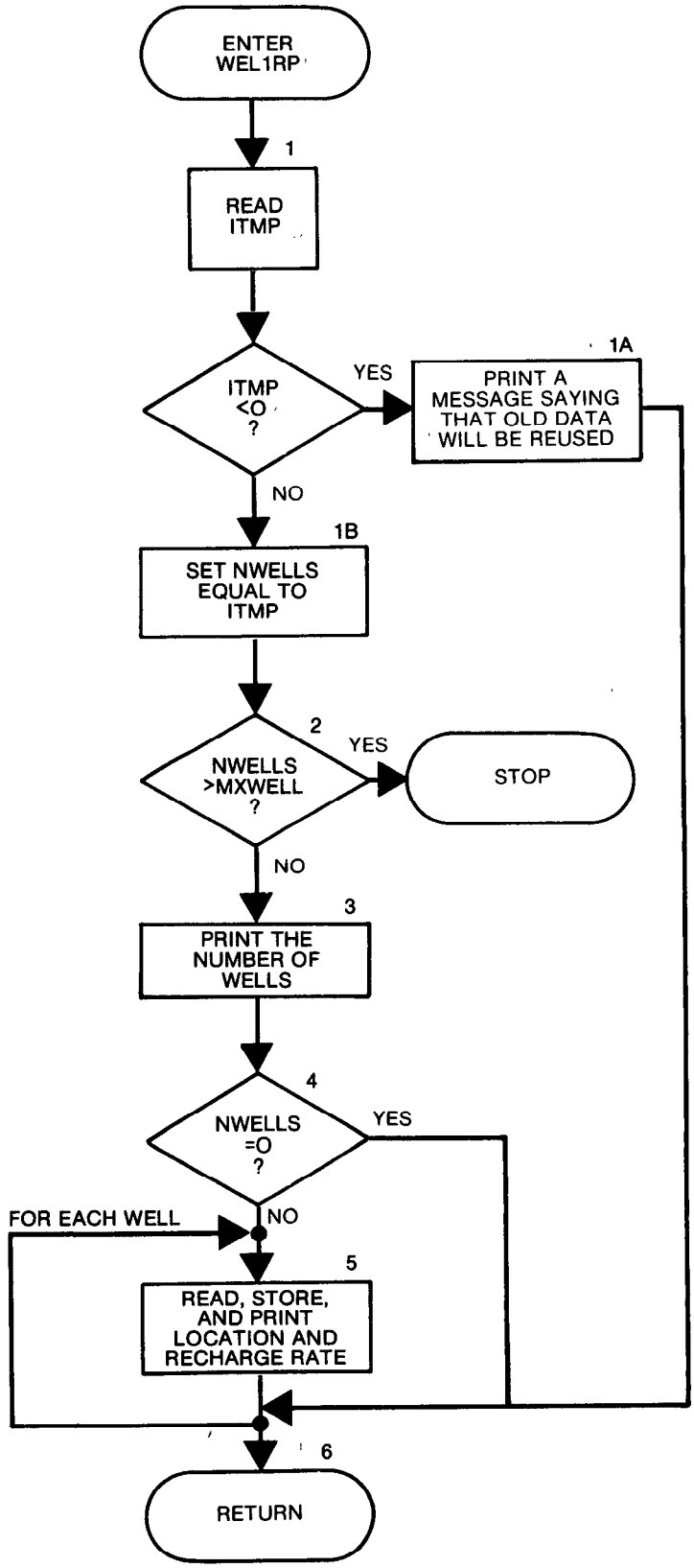
6. RETURN.

Flow Chart for Module WEL1RP

ITMP is a flag and/or the number of wells. If it is less than zero, it is a flag which indicates that the well data from the last stress period will be reused. If it is greater than or equal to zero, it is the number of wells active during the current stress period.

NWELLS is the number of wells active during the current stress period.

MXWELL is the maximum number of wells which will be active at any one time during the simulation.



```
      SUBROUTINE WEL1RP(WELL,NWELLS,MXWELL,IN,IOUT)
C
C
C-----VERSION 1544 22DEC1982 WEL1RP
C     ******************************************************************
C     READ NEW WELL LOCATIONS AND STRESS RATES
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION WELL(4,MXWELL)
C     ------------------------------------------------------------------
C
C1------READ ITMP(NUMBER OF WELLS OR FLAG SAYING REUSE WELL DATA)
      READ (IN,1) ITMP
    1 FORMAT(I10)
      IF(ITMP.GE.0) GO TO 50
C
C1A-----IF ITMP LESS THAN ZERO REUSE DATA. PRINT MESSAGE AND RETURN.
      WRITE(IOUT,6)
    6 FORMAT(1HO,'REUSING WELLS FROM LAST STRESS PERIOD')
      RETURN
C
C1B-----ITMP=>0.  SET NWELLS EQUAL TO ITMP.
   50 NWELLS=ITMP
      IF(NWELLS.LE.MXWELL) GO TO 100
C
C2------NWELLS>MXWELL.  PRINT MESSAGE. STOP.
      WRITE(IOUT,99) NWELLS,MXWELL
   99 FORMAT(1HO,'NWELLS(',I4,') IS GREATER THAN MXWELL(',I4,')')
      STOP
C
C3------PRINT NUMBER OF WELLS IN CURRENT STRESS PERIOD.
  100 WRITE (IOUT,2) NWELLS
    2 FORMAT(1HO,10X,I4,' WELLS')
C
C4------IF THERE ARE NO ACTIVE WELLS IN THIS STRESS PERIOD THEN RETURN
      IF(NWELLS.EQ.0) GO TO 260
C
C5------READ AND PRINT LAYER,ROW,COLUMN AND RECHARGE RATE.
      WRITE(IOUT,3)
    3 FORMAT(1H ,47X,'LAYER    ROW    COL    STRESS RATE   WELL NO.'/
     1,48X,45('-'))
      DO 250 II=1,NWELLS
      READ (IN,4) K,I,J,Q
    4 FORMAT(3I10,F10.0)
      WRITE (IOUT,5) K,I,J,Q,II
    5 FORMAT(48X,I3,I8,I7,G16.5,I8)
      WELL(1,II)=K
      WELL(2,II)=I
      WELL(3,II)=J
      WELL(4,II)=Q
  250 CONTINUE
C
C6------RETURN
  260 RETURN
      END
```

List of Variables for Module WEL1RP

Variable	Range	Definition
I	Module	Row number of cell containing well.
II	Module	Index for wells.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of wells. $\geq$ 0, number of wells active during the current stress period. < 0, same wells active during the last stress period will be active during the current stress period.
J	Module	Column number of cell containing well.
K	Module	Layer number of cell containing well.
MXWELL	Package	Maximum number of wells active at any one time.
NWELLS	Package	Number of wells active during the current stress period.
Q	Module	Rate at which the well adds water to the aquifer (negative for discharging well).
WELL	Package	DIMENSION (4,MXWELL), For each well: layer, row, column, and recharge rate of the well.

Narrative for Module WEL1FM

This module adds terms representing well recharge to the accumulator in which the term RHS is formulated.

1. If NWELLS is less than or equal to zero in the current stress period, there are no wells. RETURN.

2. For each well in the WELL list:

   (a) If the cell containing the well is external (IBOUND (IC,IR,IL) $\leq$ 0), bypass processing on this well and go on to the next well.

   (b) If the cell containing the well is active, subtract the value of Q from the accumulator RHS for that cell.

3. RETURN.

```
      SUBROUTINE WEL1FM(NWELLS,MXWELL,RHS,WELL,IBOUND,
     1         NCOL,NROW,NLAY)
C
C-----VERSION 1233 12MAY1987 WEL1FM
C
C     ******************************************************************
C     SUBTRACT Q FROM RHS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION RHS(NCOL,NROW,NLAY),WELL(4,MXWELL),
     1             IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C1------IF NUMBER OF WELLS <= 0 THEN RETURN.
      IF(NWELLS.LE.0) RETURN
C
C2------PROCESS EACH WELL IN THE WELL LIST.
      DO 100 L=1,NWELLS
      IR=WELL(2,L)
      IC=WELL(3,L)
      IL=WELL(1,L)
      Q=WELL(4,L)
C
C2A-----IF THE CELL IS INACTIVE THEN BYPASS PROCESSING.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C2B-----IF THE CELL IS VARIABLE HEAD THEN SUBTRACT Q FROM
C       THE RHS ACCUMULATOR.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-Q
  100 CONTINUE
C
C3------RETURN
      RETURN
      END
```

List of Variables for Module WEL1FM

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.
L	Module	Index for wells.
MXWELL	Package	Maximum number of wells active at any one time.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NWELLS	Package	Number of wells active during the current stress period.
Q	Module	Rate at which the well adds water to the aquifer (negative for discharging wells).
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.
WELL	Package	DIMENSION (4,MXWELL), For each well: layer, row, column, and recharge rate of the well.

Narrative for Module WEL1BD

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

1. Clear the rate accumulators RATIN and RATOUT and the flag (IBD) which indicates that cell-by-cell flow terms should be recorded on a disk.

2. If there are no wells, skip down to step 7.

3. Determine if the cell-by-cell flow terms for wells will be written on a disk. They will be if (1) this is the proper time step (ICBCFL is not equal to zero), (2) if the channel for well-budget terms (IWELCB) is greater than zero, and (3) if the number of wells (NWELLS) is greater than zero.

4. If budget terms are to be written on a disk, set IBD = 1 and clear the buffer (BUFF) in which they will be accumulated.

5. If the number of wells in the current stress period (NWELLS) is not equal to zero, then for each cell in the well list:

   (a) If the cell containing the well is external (IBOUND(I,J,K) $\leq$ 0), bypass further processing of the cell.

   (b) If the user has requested that cell-by-cell rates be printed (IWELCB < 0 and ICBCFL $\neq$ 0), print the rate (Q).

   (c) If the budget terms are to be saved on a disk, add the flow rate (Q) to the buffer (BUFF).

   (d) If Q is positive, add it to RATIN.

   (e) If Q is negative, add it to RATOUT.

6. If the cell-by-cell flow terms are to be recorded, call module UBUDSV to write the contents of buffer (BUFF) onto the disk.

7. Move RATIN and RATOUT into the VBVL array for printing by BAS1OT.

8. Add RATIN and RATOUT multiplied by the time-step length to the volume accumulators in the VBVL array for printing by BAS1OT.

9. Move the well budget term labels to VBNM for printing by BAS1OT.

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

11. RETURN.

Flow Chart for Module WEL1BD

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

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

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

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

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

Q is the rate at which the well recharges the aquifer. A discharging well is represented by a negative rate.

IWELCB is a flag and a unit number.

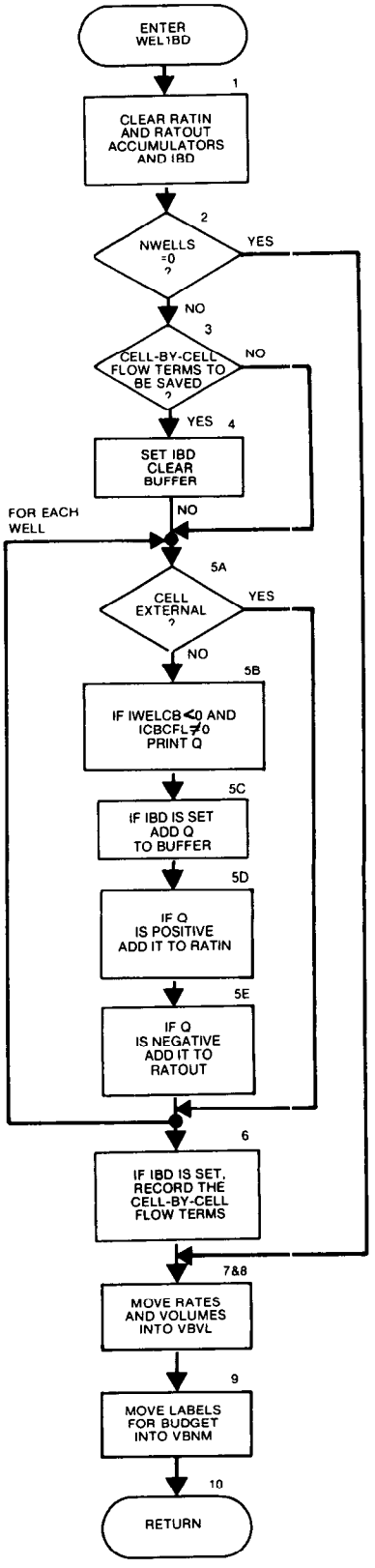
If IWELCB $>$ 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

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

If IWELCB $<$ 0, well recharge rate will be printed whenever ICBCFL is set.

ICBCFL is a flag.

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



```
      SUBROUTINE WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,WELL,IBOUND,DELT,
     1              NCOL,NROW,NLAY,KSTP,KPER,IWELCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1509 12MAY1987 WEL1BD
C     ******************************************************************
C     CALCULATE VOLUMETRIC BUDGET FOR WELLS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      DIMENSION VBNM(4,MSUM),VBVL(4,MSUM),WELL(4,MXWELL),
     1          IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
      DIMENSION TEXT(4)
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','   W','ELLS'/
C     ------------------------------------------------------------------
C
C1------CLEAR RATIN AND RATOUT ACCUMULATORS.
      RATIN=0.
      RATOUT=0.
      IBD=0
C
C2------IF THERE ARE NO WELLS DO NOT ACCUMULATE FLOW
      IF(NWELLS.EQ.0) GO TO 200
C
C3------TEST TO SEE IF CELL-BY-CELL FLOW TERMS WILL BE RECORDED.
      IF(ICBCFL.EQ.0 .OR. IWELCB.LE.0) GO TO 60
C
C4------IF CELL-BY-CELL FLOWS WILL BE SAVED THEN CLEAR THE BUFFER.
      IBD=1
      DO 50 IL=1,NLAY
      DO 50 IR=1,NROW
      DO 50 IC=1,NCOL
      BUFF(IC,IR,IL)=0.
   50 CONTINUE
C
C5------PROCESS WELLS ONE AT A TIME.
   60 DO 100 L=1,NWELLS
      IR=WELL(2,L)
      IC=WELL(3,L)
      IL=WELL(1,L)
      Q=WELL(4,L)
C
C5A-----IF THE CELL IS EXTERNAL IGNORE IT.
```

```
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 100
C
C5B-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IWELCB<0).
      IF(IWELCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
     1    KPER,KSTP,L,IL,IR,IC,Q
  900 FORMAT(1H0,4A4,'   PERIOD',I3,'   STEP',I3,'   WELL',I4,
     1    '   LAYER',I3,'   ROW ',I4,'   COL',I4,'   RATE',G15.7)
C
C5C-----IF CELL-BY-CELL FLOWS ARE TO BE SAVED THEN ADD THEM TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+Q
      IF(Q) 90,100,80
C
C5D-----PUMPING RATE IS POSITIVE(RECHARGE). ADD IT TO RATIN.
   80 RATIN=RATIN+Q
      GO TO 100
C
C5E-----PUMPING RATE IS NEGATIVE(DISCHARGE). ADD IT TO RATOUT.
   90 RATOUT=RATOUT-Q
  100 CONTINUE
C
C6------IF CELL-BY-CELL FLOWS WILL BE SAVED CALL UBUDSV TO RECORD THEM
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IWELCB,BUFF,NCOL,NROW,
     1                         NLAY,IOUT)
C
C7------MOVE RATES INTO VBVL FOR PRINTING BY MODULE BAS1OT.
  200 VBVL(3,MSUM)=RATIN
      VBVL(4,MSUM)=RATOUT
C
C8------MOVE RATES TIMES TIME STEP LENGTH INTO VBVL ACCUMULATORS.
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
C
C9------MOVE BUDGET TERM LABELS INTO VBNM FOR PRINTING.
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C10-----INCREMENT BUDGET TERM COUNTER(MSUM).
      MSUM=MSUM+1
C
C11-----RETURN
      RETURN
      END
```

List of Variables for Module WEL1BD

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.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be either printed or recorded (depending on IWELCB) for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
IWELCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, well recharge rate will be printed whenever ICBCFL is set.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for wells.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXWELL	Package	Maximum number of wells active at any one time.

List of Variables for Module WEL1BD (Continued)

Variable	Range	Definition
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NWELLS	Package	Number of wells active during the current stress period.
Q	Module	Rate at which the well adds water to the aquifer (negative for discharging wells).
RATIN	Module	Accumulator for the total flow into the flow field from wells.
RATOUT	Module	Accumulator for the total flow out of the flow field into wells.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N) Rate for the current time step into the flow field. (2,N) Rate for the current time step out of the flow field. (3,N) Volume into the flow field during simulation. (4,N) Volume out of the flow field during simulation.
WELL	Package	DIMENSION (4,MXWELL), For each well: layer, row, column, and recharge rate of the well.

CHAPTER 9

DRAIN PACKAGE

Conceptualization and Implementation

The Drain Package is designed to simulate the effects of features such as agricultural drains, which remove water from the aquifer at a rate proportional to the difference between the head in the aquifer and some fixed head or elevation, so long as the head in the aquifer is above that elevation, but which have no effect if head falls below that level. The discussion in this section is phrased on the assumption that the features to be simulated are actually agricultural drains.

Figure 39 shows a cross section through a cell, illustrating concepts underlying the simulation of drains in the model. The drain is assumed to run only partially full, so that head within the drain is approximately equal to the median drain elevation, $d_{i,j,k}$. The head computed by the model for cell i,j,k ($h_{i,j,k}$) is actually an average value for the cell, and is normally assumed to prevail at some distance from the drain itself. The drain head, $d_{i,j,k}$ prevails only locally, within the drain--it does not characterize the cell as a whole. Between the drain and the area in which head $h_{i,j,k}$ prevails there exists a radial or semiradial flow pattern in the vertical plane, normally characterized by progressively steeper head gradients as the drain is approached. The head loss within this converging flow pattern forms one part of the head difference $h_{i,j,k}-d_{i,j,k}$. An additional component of head loss may occur in the immediate vicinity of the drain if the hydraulic conductivity in that region differs from the average value used for cell i,j,k--because of the presence of foreign material around

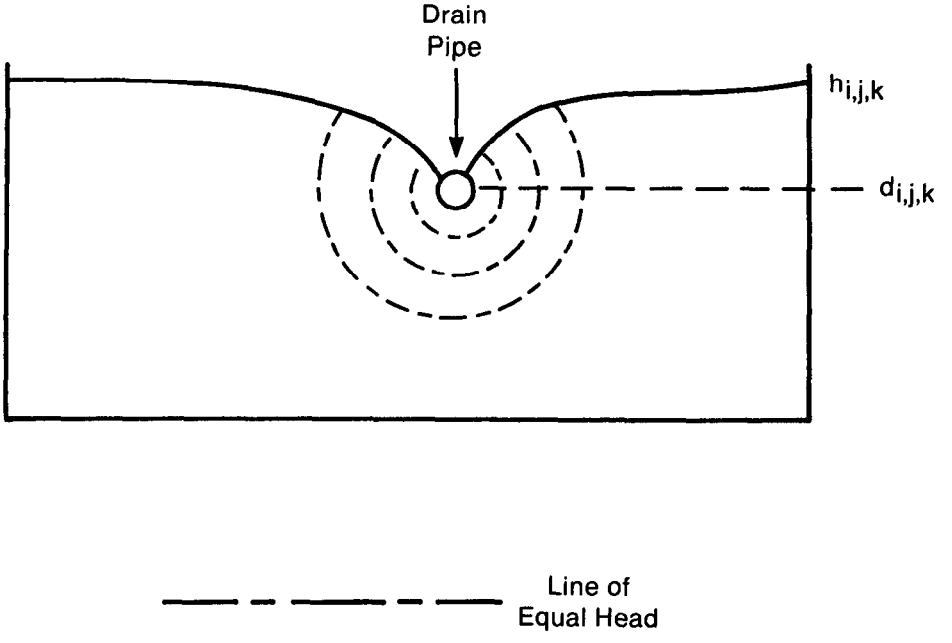


Figure 39.—Cross section through cell i,j,k illustrating head loss in convergent flow into drain.

the drain pipe, or channel-bed material in the case of an open drain (figure 40). Finally, head losses occur through the wall of a drain pipe, depending upon the number and size of the openings in the pipe, and the degree to which those openings may be blocked by chemical precipitates, plant roots, etc.

The three processes discussed above--convergent flow toward the drain, flow through material of different conductivity immediately around the drain, and flow through the wall of the drain--each generate head losses which may be assumed proportional to the discharge, QD, through the system--that is, the discharge from cell i,j,k into the drain. Because these head losses occur in series, the total head loss $h_{i,j,k}-d_{i,j,k}$ may also be taken as proportional to QD. This has been done in the method of simulation embodied in the Drain Package. That is, it has been assumed that the drain function is described by the equation pair

$$QD_{i,j,k} = CD_{i,j,k}(h_{i,j,k} - d_{i,j,k}) \qquad \text{for } h_{i,j,k} > d_{i,j,k} \qquad (69)$$

$$QD_{i,j,k} = 0 \qquad \text{for } h_{i,j,k} \leq d_{i,j,k} \qquad (70)$$

The coefficient $CD_{i,j,k}$ of equation (69) is a lumped (or equivalent) conductance describing all of the head loss between the drain and the region of cell i,j,k in which the head $h_{i,j,k}$ can be the assumed to prevail. It depends on the characteristics of the convergent flow pattern toward the drain, as well as on the characteristics of the drain itself and its immediate environment.

One could attempt to calculate values for CD by developing approximate equations for conductance for the three flow processes, and then calculate the equivalent series conductance. The conductance for each process would be

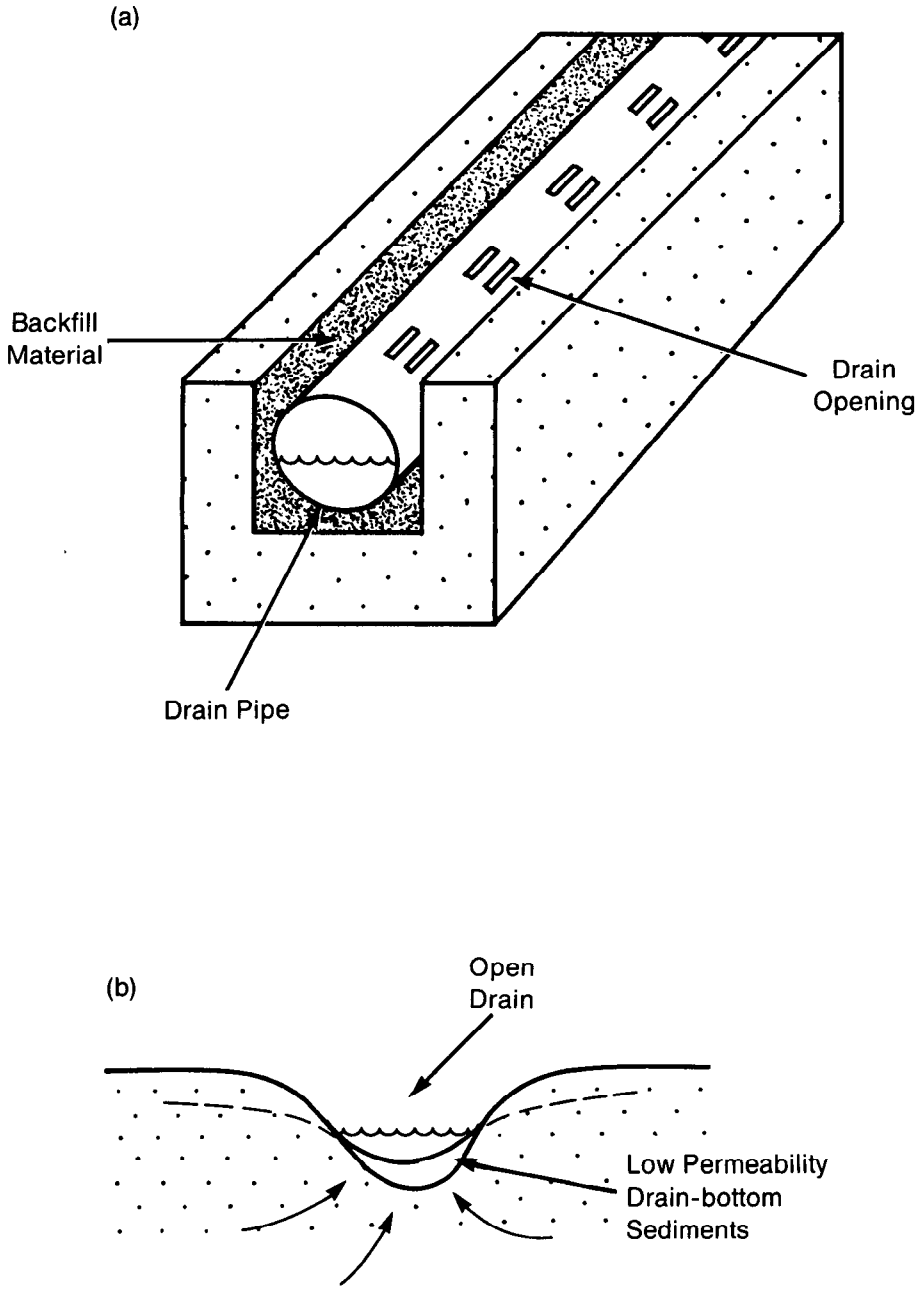


Figure 40.—Factors affecting head loss immediately around a drain: (a) buried drain pipe in backfilled ditch and (b) open drain.

based on the formulation of a one-dimensional flow equation. The formulations vary significantly depending on the specific drain system being simulated, so no general formulation for calculating CD is presented here. Also, in most situations a specific formulation would require detailed information that is not usually available, such as detailed head distribution around the drain, aquifer hydraulic conductivity near the drain, distribution of the fill material, hydraulic conductivity of fill material, number and size of the drain pipe openings, the amount of clogging materials, and the hydraulic conductivity of the clogging materials. In practice, it is more common to calculate CD from measured values of QD and h-d using equation (69). If h-d is not accurately known, CD is usually adjusted during model calibration in order to match measured values of QD to model calculated values.

Figure 41 shows a graph of QD vs. $h_{i,j,k}$ as defined by equations (69) and (70); the function is similar to that for flow between a surface stream and the aquifer (figure 36) except that flow into the aquifer is excluded, and positive values of QD have been taken as corresponding to flow into the drain. With proper selection of coefficients, the River Package could in fact be utilized to perform the functions of the Drain Package.

Because $QD_{i,j,k}$ in equation (69) has been taken as a flow out of cell i,j,k, it must be subtracted from the left side of equation (24) for each cell affected by a drain, provided the head $h_{i,j,k}$ is above the drain elevation. This is accomplished in the Drain Package by testing to determine whether head exceeds drain elevation, and if so, by adding the term $-CD_{i,j,k}$ to $HCOF_{i,j,k}$ (equation (26)) and adding the term $-CD_{i,j,k}d_{i,j,k}$ to $RHS_{i,j,k}$, as the matrix equations are assembled.

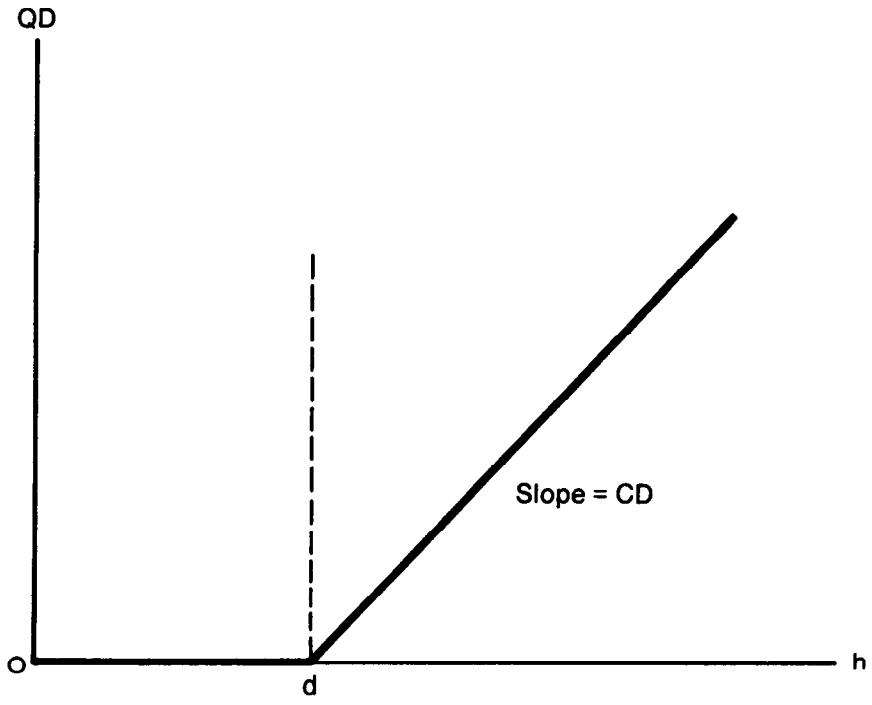


Figure 41.—Plot of flow, QD, into a drain as a function of head, h, in a cell where the elevation of the drain is d and the conductance is CD.

Drain Package Input

Input to the Drain (DRN) Package is read from the unit specified in IUNIT(3).

FOR EACH SIMULATION

DRN1AL

1.	Data:	MXDRN	IDRNCB
	Format:	I10	I10

FOR EACH STRESS PERIOD

DRN1RP

2.	Data:	ITMP				
	Format:	I10				
3.	Data:	Layer	Row	Col	Elevation	Cond
	Format:	I10	I10	I10	F10.0	F10.0

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

Explanation of Fields Used in Input Instructions

MXDRN--is the maximum number of drain cells active at one time.

IDRNCB--is a flag and a unit number.

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

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

If IDRNCB < 0, drain leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

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

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

Layer--is the layer number of the cell containing the drain.

Row--is the row number of the cell containing the drain.

Column--is the column number of the cell containing the drain.

Elevation--is elevation of the drain.

Cond--is the hydraulic conductance of the interface between the aquifer and the drain.

SAMPLE INPUT TO THE DRAIN PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS				
1	{MXDRNR, IDRNCB}	3	55			
2	{ITMP} FOR FIRST STRESS PERIOD	3				
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR FIRST DRAIN	2	6	4	220.	.7
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR SECOND DRAIN	2	7	4	225.	.9
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR THIRD DRAIN	2	5	4	210.	.8
2	{ITMP} FOR SECOND STRESS PERIOD	-1				
2	{ITMP} FOR THIRD STRESS PERIOD	-1				
2	{ITMP} FOR FOURTH STRESS PERIOD	2				
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR FIRST DRAIN	2	5	4	210.	.8
3	{LAYER, ROW, COLUMN, ELEVATION, COND } FOR SECOND DRAIN	2	6	4	220.	.7
2	{ITMP} FOR FIFTH STRESS PERIOD	0				
2	{ITMP} FOR SIXTH STRESS PERIOD	-1				

Module Documentation for the Drain Package

The Drain Package (DRN1) consists of four modules, all of which are called by the MAIN program. The modules are:

DRN1AL — Allocates space for an array that contains the drain list (DRAI).

DRN1RP — Reads location, drain elevation, and drain conductance of each cell containing a drain.

DRN1FM — Adds the terms $-CD_{i,j,k}$ and $-CD_{i,j,k}d_{i,j,k}$ to the accumulators $HCOF_{i,j,k}$ and $RHS_{i,j,k}$, respectively.

DRN1BD — Calculates the rates and accumulated volume of drainage from the flow system.

Narrative for Module DRN1AL

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

1. Print a message identifying the package and initialize NDRAIN (number of drains).

2. Read and print MXDRAN (the maximum number of drains) and IDRNCB (the file number for saving cell-by-cell flow terms or a flag indicating that cell-by-cell flow terms should be printed).

3. Set LCDRAI (which will point to the first element in the drain list) equal to ISUM (which points to the first unallocated element in the X array.

4. Calculate the amount of space needed for the drain list (five values for each drain--row, column, layer, drain elevation, and drain conductance).

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

6. RETURN.

Flow Chart for Module DRN1AL

NDRAIN is the number of drains being simulated at any given time.

MXDRN is the maximum number of drains simulated.

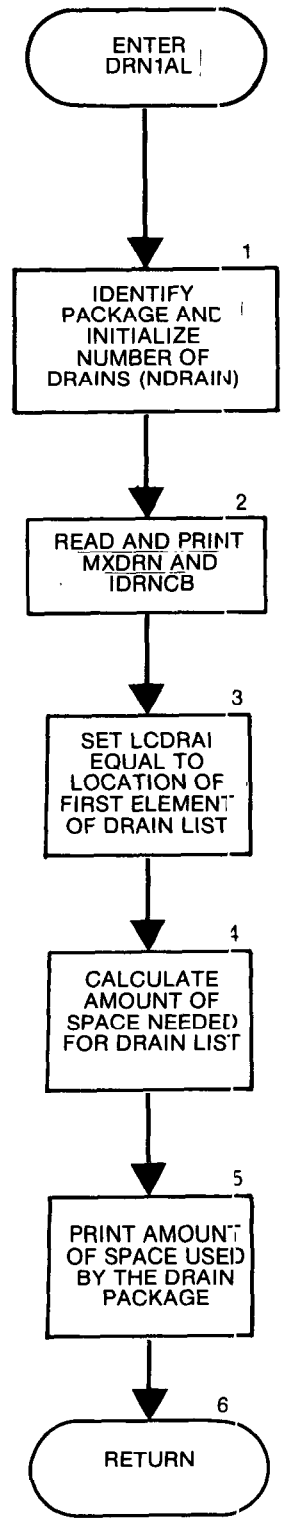
IDRNCB is a flag and a unit number.

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

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

If IDRNCB < 0, drain leakage for each drain will be printed whenever ICBCFL is set.

LCDRAI is the location, in the X array, of the list of drain data (DRAI).



```
      SUBROUTINE DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,IN,IOUT,
     1                  IDRNCB)
C
C-----VERSION 1604 12MAY1987 DRN1AL
C     ***************************************************************
C     ALLOCATE ARRAY STORAGE FOR DRAIN PACKAGE
C     ***************************************************************
C
C        SPECIFICATIONS:
C     ---------------------------------------------------------------
C     ---------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE NDRAIN.
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'DRN1 -- DRAIN PACKAGE, VERSION 1, 9/1/87',
     1' INPUT READ FROM UNIT',I3)
      NDRAIN=0
C
C2------READ & PRINT MXDRN & IDRNCB(UNIT & FLAG FOR CELL-BY-CELL FLOW)
      READ(IN,2) MXDRN,IDRNCB
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXDRN
    3 FORMAT(1H ,'MAXIMUM OF',I5,' DRAINS')
      IF(IDRNCB.GT.0) WRITE(IOUT,9) IDRNCB
    9 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE RECORDED ON UNIT',I3)
      IF(IDRNCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOWS WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3------SET LCDRAI EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCDRAI=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY THE DRAIN PACKAGE.
      ISP=5*MXDRN
      ISUM=ISUM+ISP
C
C5------PRINT AMOUNT OF SPACE USED BY DRAIN PACKAGE.
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED FOR DRAINS')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6------RETURN
      RETURN
      END
```

List of Variables for Module DRN1AL

Variable	Range	Definition
IDRNCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will be neither printed nor recorded. < 0, leakage for each drain will be printed.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISP	Module	Number of words in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM - 1.
LCDRAI	Package	Location in the X array of the first element of array DRAI.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN Program.
MXDRN	Package	Maximum number of drains active at any one time.
NDRAIN	Package	Number of drains active during the current stress period.

Narrative for Module DRN1RP

This module reads data to build the drain list.

1. Read ITMP. ITMP is the number of drains or a flag indicating that drain data from the previous stress period should be reused.

2. Test ITMP. If ITMP is less than zero, the drain data read for the last stress period will be reused. Print a message to that effect and RETURN.

3. If ITMP is greater than or equal to zero, it is the number of drains for this stress period. Set the number of drains (NDRAIN) in the current stress period equal to ITMP.

4. Compare the number of drains (NDRAIN) in the current stress period to the number specified as the maximum for the simulation (MXDRN). If NDRAIN is greater than MXDRN, STOP.

5. Print the number of drains in the current stress period (NDRAIN).

6. See if there are any drains. If there are no drains in the current stress period (NDRAIN = 0), bypass further drain processing.

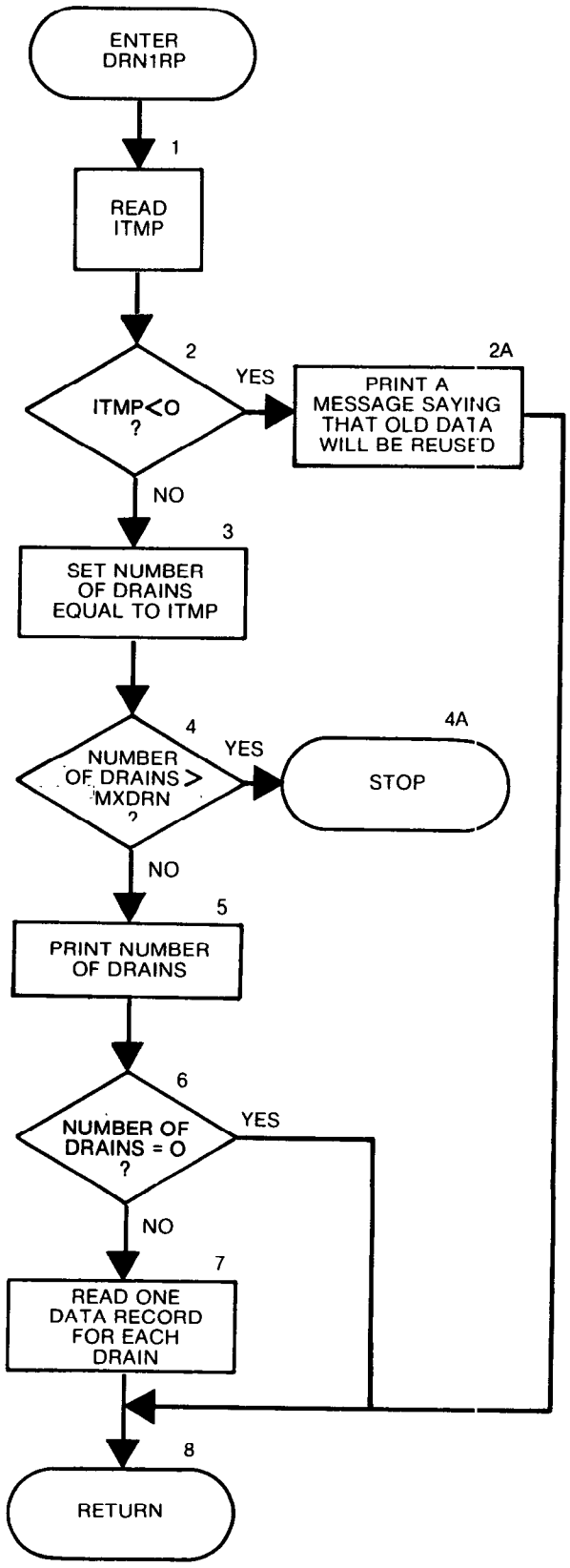
7. Read and print the layer, row, column, elevation, and conductance for each drain.

8. RETURN.

Flow Chart for Module DRN1RP

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

MXDRN is the maximum number of drains to be simulated.



```
      SUBROUTINE DRN1RP(DRAI,NDRAIN,MXDRN,IN,IOUT)
C
C
C-----VERSION 1603 25APR1983 DRN1RP
C     ******************************************************************
C     READ DRAIN LOCATIONS, ELEVATIONS, AND CONDUCTANCES
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION DRAI(5,MXDRN)
C     ------------------------------------------------------------------
C
C1------READ ITMP(NUMBER OF DRAIN CELLS OR FLAG TO REUSE DATA)
      READ(IN,8) ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
    7 FORMAT(1H0,'REUSING DRAINS FROM LAST STRESS PERIOD')
      RETURN
C
C3------IF ITMP=>0 THEN IT IS THE NUMBER OF DRAINS.
   50 NDRAIN=ITMP
      IF(NDRAIN.LE.MXDRN) GO TO 100
C
C4------IF NDRAIN>MXDRN THEN STOP
      WRITE(IOUT,99) NDRAIN,MXDRN
   99 FORMAT(1H0,'NDRAIN(',I4,') IS GREATER THAN MXDRN(',I4,')')
      STOP
C
C5------PRINT NUMBER OF DRAINS IN THIS STRESS PERIOD.
  100 WRITE(IOUT,1) NDRAIN
    1 FORMAT(1H0,//1X,I5,' DRAINS')
C
C6------IF THERE ARE NO DRAINS THEN RETURN.
      IF(NDRAIN.EQ.0) GO TO 260
C
C7------READ AND PRINT DATA FOR EACH DRAIN.
      WRITE(IOUT,3)
    3 FORMAT(1H0,15X,'LAYER',5X,'ROW',5X
     1,'COL   ELEVATION   CONDUCTANCE   DRAIN NO.'/1X,15X,60('-'))
      DO 250 II=1,NDRAIN
      READ (IN,4) K,I,J,DRAI(4,II),DRAI(5,II)
    4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,DRAI(4,II),DRAI(5,II),II
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,I8)
      DRAI(1,II)=K
      DRAI(2,II)=I
      DRAI(3,II)=J
  250 CONTINUE
C
C8------RETURN
  260 RETURN
C
      END
```

List of Variables for Module DRN1RP

Variable	Range	Definition
DRAI	Package	DIMENSION (5,MXDRN), For each drain: layer, row, column, head in drain, and conductance into drain.
I	Module	Index for rows.
II	Module	Index for drains.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of drains. $\geq$ 0, number of drains active during the current stress period. < 0, same drains active during the last stress period will be active during the current stress period.
J	Module	Index for columns.
K	Module	Index for layers.
MXDRN	Package	Maximum number of drains active at any one time.
NDRAIN	Package	Number of drains active during the current stress period.

Narrative for Module DRN1FM

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

1. If NDRAIN is less than or equal to zero in the current stress period, there are no drains. RETURN.

2. For each drain in the drain list, DO STEPS 3-7.

3. Determine the column (IC), row (IR), and layer (IL).

4. If the cell is external (IBOUND(IC, IR, IL) $\leq$ 0), bypass processing on this drain and go on to the next drain.

5. If the cell is internal, get the drain data (elevation and conductance).

6. If the head in the aquifer (HHNEW) is greater than the elevation of the drain, there is no drain leakage. RETURN.

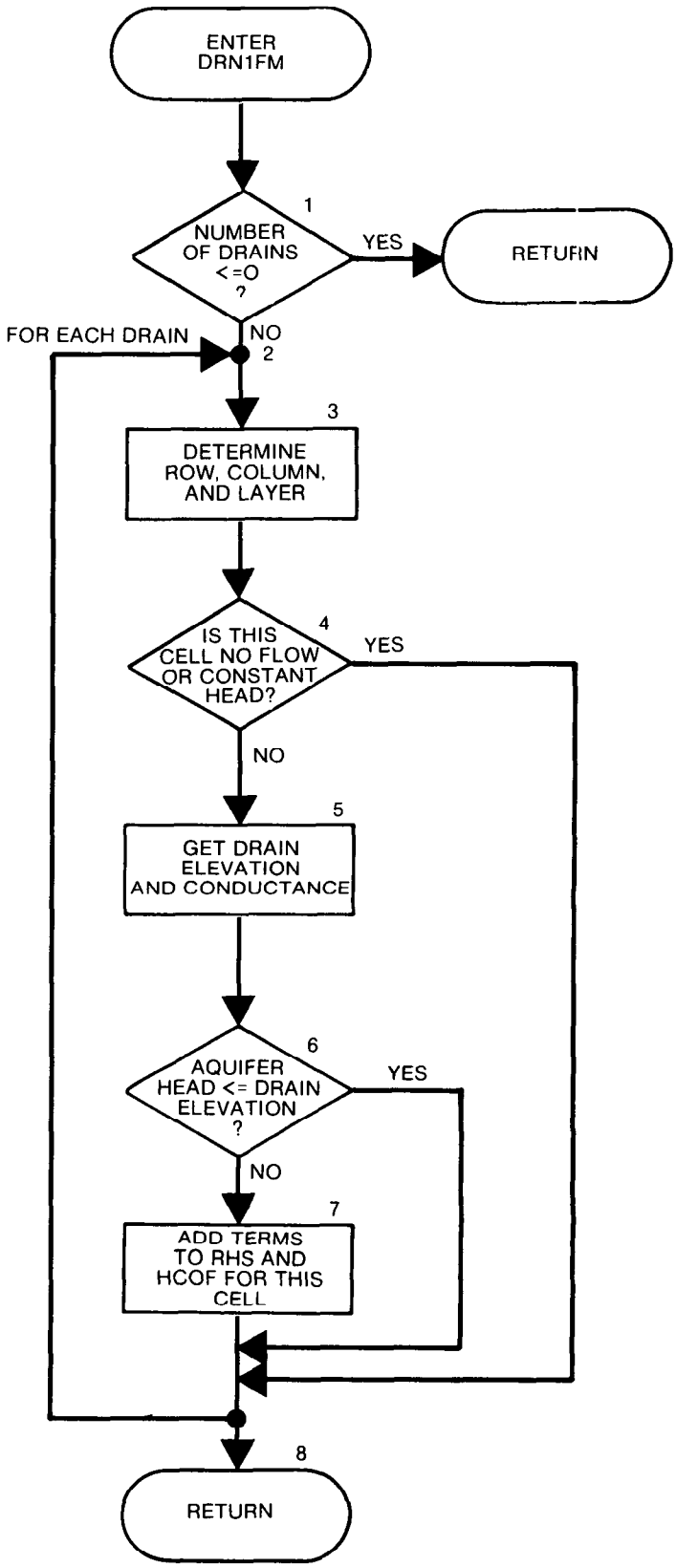
7. If the head in the aquifer (HHNEW) is greater than the elevation of the drain (EL), add the term -C*EL (C is the drain conductance) to the accumulator RHS and the term -C to the accumulator HCOF.

8. RETURN.

Flow Chart for Module DRN1FM

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

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



```
      SUBROUTINE DRN1FM(NDRAIN,MXDRN,DRAI,HNEW,HCOF,RHS,IBOUND,
     1                  NCOL,NROW,NLAY)
C
C-----VERSION 1030 10APR1985 DRN1FM
C
C     ******************************************************************
C     ADD DRAIN FLOW TO SOURCE TERM
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION DRAI(5,MXDRN),HNEW(NCOL,NROW,NLAY),
     1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     1          HCOF(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NDRAIN<=0 THERE ARE NO DRAINS. RETURN
      IF(NDRAIN.LE.0) RETURN
C
C2------PROCESS EACH CELL IN THE DRAIN LIST
      DO 100 L=1,NDRAIN
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING DRAIN.
      IL=DRAI(1,L)
      IR=DRAI(2,L)
      IC=DRAI(3,L)
C
C4-------IF THE CELL IS EXTERNAL SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C5-------IF THE CELL IS INTERNAL GET THE DRAIN DATA.
      EL=DRAI(4,L)
      HHNEW=HNEW(IC,IR,IL)
C
C6------IF HEAD IS LOWER THAN DRAIN THEN SKIP THIS CELL.
      IF(HHNEW.LE.EL) GO TO 100
C
C7------HEAD IS HIGHER THAN DRAIN. ADD TERMS TO RHS AND HCOF.
      C=DRAI(5,L)
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-C*EL
  100 CONTINUE
C
C8------RETURN
      RETURN
      END
```

List of Variables for Module DRN1FM

Variable	Range	Definition
C	Module	Conductance into the drain.
DRAI	Package	DIMENSION (5,MXDRN), For each drain: layer, row, column, head in the drain and conductance into the drain.
EL	Module	Elevation of the drain (head in the drain).
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HHNEW	Module	Head in the cell containing the drain.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
MXDRN	Package	Maximum number of drains active at any one time.
NCOL	Global	Number of columns in the grid.
NDRAIN	Package	Number of drains active during the current stress period.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

Narrative for Module DRN1BD

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

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

2. If there are no drains (NDRAIN $\leq$ 0), skip down to step 12 and put zeros into the budget terms for drains.

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

4. For each drain, do steps 3-11 accumulating flows into drains.

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

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

7. Get the drain parameters from the drain list.

8. If the head in the cell is less than the elevation of the drain, bypass further processing of this drain.

9. If the head in the cell is greater than the elevation of the drain, set "Q" equal to the conductance of the drain (C) times the drain elevation (EL) minus the head in the cell (HHNEW) (Q = C*(EL - HHNEW)). Add Q to the accumulator RATOUT to get the total flow from the aquifer into drains.

10. If the cell-by-cell flow terms are to be printed (IDRNCB < 0 and ICBCFL ≠ 0), print Q.

11. If the cell-by-cell flow terms for drains are to be saved, add Q to the buffer (BUFF).

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

13. Move RATOUT into the VBVL array for printing by BAS1OT. Add RATOUT multiplied by the time-step length to the volume accumulator in VBVL for printing by BAS1OT. Move the drain budget-term labels to VBNM for print by BAS1OT.

14. Increment the budget-term counter (MSUM). See the section in the Basic Package for a detailed explanation of VBVL, VBNM, and MSUM.

15. RETURN.

Flow Chart for Module DRN1BD

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

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

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

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

Q is the negative of discharge to a drain.

EL is the elevation of the drain.

IDRNCB is a flag and a unit number.

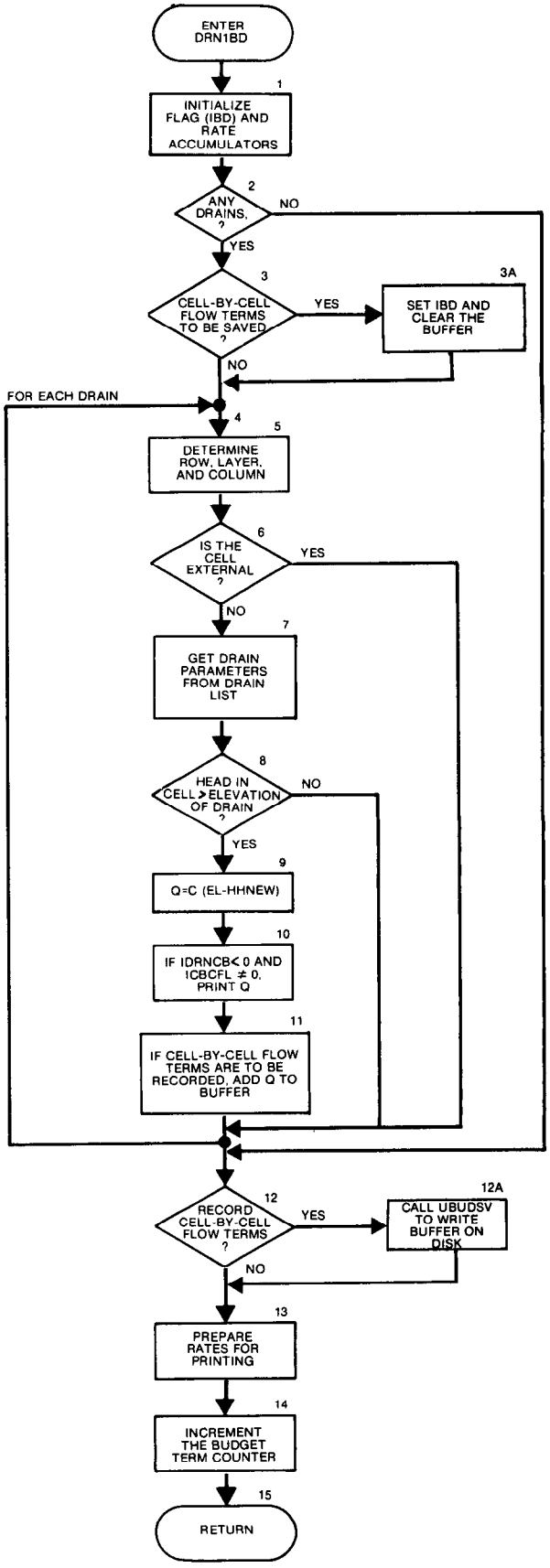
If IDRNCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

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

If IDRNCB < 0, drain leakage for each drain will be printed whenever ICBCFL is set.

ICBCFL is a flag.

If ICBCFL ≠ 0, cell-by-cell flow terms will be either printed or recorded for the current time step.



```
      SUBROUTINE DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,DRAI,DELT,HNEW,
     1        NCOL,NROW,NLAY,IBOUND,KSTP,KPER,IDRNCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1338 22AUG1987 DRN1BD
C
C     ******************************************************************
C     CALCULATE VOLUMETRIC BUDGET FOR DRAINS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      DOUBLE PRECISION HNEW
C
      DIMENSION VBNM(4,MSUM),VBVL(4,MSUM),DRAI(5,MXDRN),
     1           HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     2           BUFF(NCOL,NROW,NLAY)
      DIMENSION TEXT(4)
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','  DR','AINS'/
C     ------------------------------------------------------------------
C
C1------INITIALIZE CELL-BY-CELL FLOW TERM FLAG (IBD) AND
C1------ACCUMULATORS (RATIN AND RATOUT).
      RATOUT=0.
      IBD=0
C
C2------IF THERE ARE NO DRAINS THEN DO NOT ACCUMULATE DRAIN FLOW
      IF(NDRAIN.LE.0) GO TO 200
C
C3------TEST TO SEE IF CELL-BY-CELL FLOW TERMS ARE NEEDED.
      IF(ICBCFL.EQ.0 .OR. IDRNCB.LE.0) GO TO 60
C
C3B-----CELL-BY-CELL FLOW TERMS ARE NEEDED SET IBD AND CLEAR BUFFER.
      IBD=1
      DO 50 IL=1,NLAY
      DO 50 IR=1,NROW
      DO 50 IC=1,NCOL
      BUFF(IC,IR,IL)=0.
   50 CONTINUE
C
C4------FOR EACH DRAIN ACCUMULATE DRAIN FLOW
   60 DO 100 L=1,NDRAIN
C
C5------GET LAYER, ROW & COLUMN OF CELL CONTAINING REACH.
      IL=DRAI(1,L)
      IR=DRAI(2,L)
```

```
      IC=DRAI(3,L)
C
C6------IF CELL IS EXTERNAL IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C7------GET DRAIN PARAMETERS FROM DRAIN LIST.
      EL=DRAI(4,L)
      C=DRAI(5,L)
      HHNEW=HNEW(IC,IR,IL)
C
C8------IF HEAD LOWER THAN DRAIN THEN FORGET THIS CELL.
      IF(HHNEW.LE.EL) GO TO 100
C
C9------HEAD HIGHER THAN DRAIN.  CALCULATE Q=C*(EL-HHNEW).
C9------SUBTRACT Q FROM RATOUT.
      Q=C*(EL-HHNEW)
      RATOUT=RATOUT-Q
C
C10-----PRINT THE INDIVIDUAL RATES IF REQUESTED(IDRNCB<0).
      IF(IDRNCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
     1    KPER,KSTP,L,IL,IR,IC,Q
  900 FORMAT(1H0,4A4,'   PERIOD',I3,'   STEP',I3,'   DRAIN',I4,
     1    '   LAYER',I3,'   ROW',I4,'   COL',I4,'   RATE',G15.7)
C
C11-----IF C-B-C FLOW TERMS ARE TO BE SAVED THEN ADD Q TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+Q
  100 CONTINUE
C
C12-----IF C-B-C FLOW TERMS WILL BE SAVED CALL UBUDSV TO RECORD THEM.
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IDRNCB,BUFF,NCOL,NROW,
     1                         NLAY,IOUT)
C
C13-----MOVE RATES,VOLUMES & LABELS INTO ARRAYS FOR PRINTING.
  200 VBVL(3,MSUM)=0.
      VBVL(4,MSUM)=RATOUT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C14-----INCREMENT BUDGET TERM COUNTER
      MSUM=MSUM+1
C
C15-----RETURN
      RETURN
      END
```

List of Variables for Module DRN1BD

Variable	Range	Definition
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
C	Module	Conductance into drains.
DELT	Global	Length of the current time step.
DRAI	Package	DIMENSION (5,MXDRN), For each drain: layer, row, column, head in the drain and conductance into the drain.
EL	Module	Elevation of the drain (head in the drain).
HHNEW	Module	Head in the cell containing the drain.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBD	Package	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be recorded for the current time step.
IDRNCB	Package	Flag. > 0 and if ICBCFL ≠ 0, cell-by-cell flow terms for the DRN1 Package will be recorded on UNIT = IDRNCB.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
KPER	Global	Stress period counter.

List of Variables for Module DRN1BD (Continued)

Variable	Range	Definition
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for drains.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXDRN	Package	Maximum number of drains active at any one time.
NCOL	Global	Number of columns in the grid.
NDRAIN	Package	Number of drains active during the current stress period.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
Q	Module	Flow from a drain into a cell. (Reverse the sign to get the flow into the drain.)
RATOUT	Module	Accumulator for the total flow out of the flow field into the drains.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N) Rate for the current time step into the flow field. (2,N) Rate for the current time step out of the flow field. (3,N) Volume into the flow field during simulation. (4,N) Volume out of the flow field during simulation.

CHAPTER 10

EVAPOTRANSPIRATION PACKAGE

Conceptualization and Implementation

The Evapotranspiration (ET) Package simulates the effects of plant transpiration and direct evaporation in removing water from the saturated ground water regime. The approach is based on the following assumptions: (1) when the water table is at or above a specified elevation, termed the "ET surface" in this report, evapotranspiration loss from the water table occurs at a maximum rate specified by the user; (2) when the depth of the water table below the ET surface elevation exceeds a specified interval, termed the "extinction depth" or "cutoff depth" in this report, evapotranspiration from the water table ceases; and (3) between these limits, evapotranspiration from the water table varies linearly with water table elevation.

This can be expressed in equation form as

$$R_{ETi,j} = R_{ETMi,j} \qquad h_{i,j,k} > h_{si,j} \tag{71}$$

$$R_{ETi,j} = 0 \qquad h_{i,j,k} < h_{si,j} - d_{i,j} \tag{72}$$

$$R_{ETi,j} = R_{ETMi,j}\left\{\frac{h_{i,j,k} - (h_{si,j} - d_{i,j})}{d_{i,j}}\right\} \qquad (h_{si,j} - d_{i,j}) \leq h_{i,j,k} \leq h_{si,j} \tag{73}$$

where $R_{ETi,j}$ is the rate of loss per unit surface area of water table due to evapotranspiration, in volume of water per unit area per unit time, within the map area $DELR_j DELC_i$; $h_{i,j,k}$ is the head, or water table elevation in the cell from which the evapotranspiration occurs; $R_{ETMi,j}$ is the maximum possible value of $R_{ETi,j}$; $h_{si,j}$ is the ET surface elevation, or the water table elevation at which this maximum value of evapotranspiration loss occurs; and $d_{i,j}$ is the cutoff or extinction depth, such that when the

distance between $h_{si,j}$ and $h_{i,j,k}$ exceeds $d_{i,j}$ evapotranspiration ceases.

In implementing the finite difference approach the volumetric rate of evapotranspiration loss from a given cell is required. This is given as the product of the loss rate per unit area, and the horizontal surface area, $DELR_j DELC_i$, of the cell from which the loss occurs, i.e.

$$Q_{ETi,j} = R_{ETi,j} * DELR_j * DELC_i \tag{74}$$

where $Q_{ETi,j}$ is the evapotranspiration, in volume of water per unit time, through the area $DELR_j DELC_i$. If the maximum value of $Q_{ETi,j}$ (corresponding to $R_{ETMi,j}$) is designated $Q_{ETMi,j}$, equations (71)-(73) can be expressed in terms of volumetric discharge as

$$Q_{ETi,j} = Q_{ETMi,j} \qquad h_{i,j,k} > h_{si,j} \tag{75}$$

$$Q_{ETi,j} = 0 \qquad h_{i,j,k} < h_{si,j} - d_{i,j} \tag{76}$$

$$Q_{ETi,j} = Q_{ETMi,j}\left\{\frac{h_{i,j,k} - (h_{si,j} - d_{i,j})}{d_{i,j}}\right\} \qquad (h_{si,j} - d_{i,j}) \leq h_{i,j,k} \leq h_{si,j} \tag{77}$$

Figure 42 shows a graph of evapotranspiration loss, $Q_{ETi,j}$, vs head in cell i,j,k based on equations (75)-(77). Comparison of the ET function with the river or drain functions shows that the three are mathematically similar, except that the linear portion of the ET function is bounded at both ends by constant values, rather than only at the lower end.

Evapotranspiration is drawn from only one cell in the vertical column beneath the map area $DELR_j * DELC_i$; the user designates the cell (i.e. the layer, k) using one of two options. Under the first option, evapotranspiration is always drawn from the uppermost layer of the model; under the second, the user specifies the cell, within the vertical column at i,j, from which the evapotranspiration is to be taken. In either case the computed evapotrans-

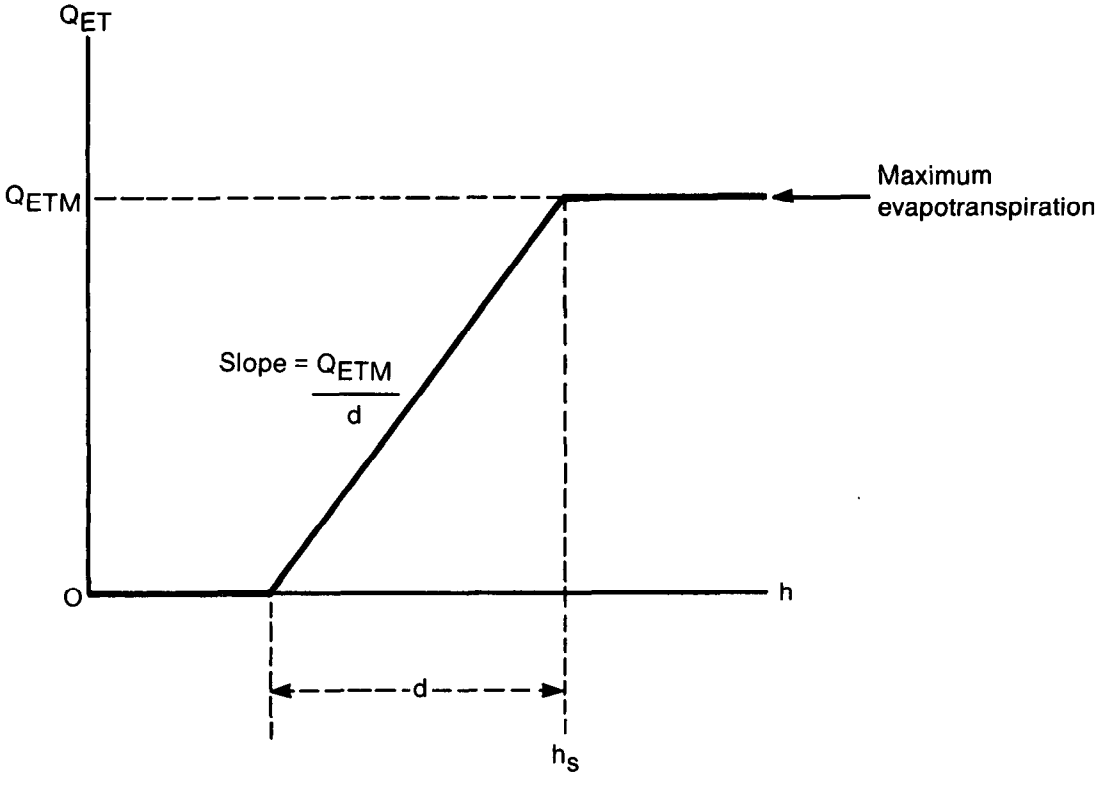


Figure 42.—Plot of volumetric evapotranspiration, Q_{ET}, as a function of head, h, in a cell where d is the cutoff depth and h_s is the ET surface elevation.

piration has no influence on the simulation if the designated cell is either a no-flow cell or a constant head cell.

For each cell location, (i, j), in the horizontal plane, and for each stress period (unless an option is exercised to use prior values) the ET package reads values of R_{ETM} (maximum evapotranspiration loss per unit area per unit time) into an array labelled EVTR. These rates are immediately multiplied by cell areas, $DELR_j * DELC_i$, to obtain the maximum volumetric rate of evapotranspiration from each cell, Q_{ETM}; these maximum volumetric rates then replace the values of $R_{ETMi,j}$ in the EVTR array. Thus, the input to the EVTR array consists of maximum evapotranspiration rates per unit area, and as such must have dimensions Lt^{-1}. In the calculation carried out within the model, however, the entries in the EVTR array appear as maximum volumetric rates, having dimensions L^3t^{-1}.

Values of $h_{si,j}$, the ET surface elevation (or water table elevation at which evapotranspiration is maximum), are read into the two dimensional array SURF by the ET package; values of the cutoff depth or extinction depth are read into the two-dimensonal array EXDP. Because the term $Q_{ETi,j}$ of equations (75)-(77) has been defined as an outflow from the aquifer it must be subtracted from the left side of equation (24). In terms of the expressions HCOF and RHS of equation (26), this is accomplished in the ET package as follows:

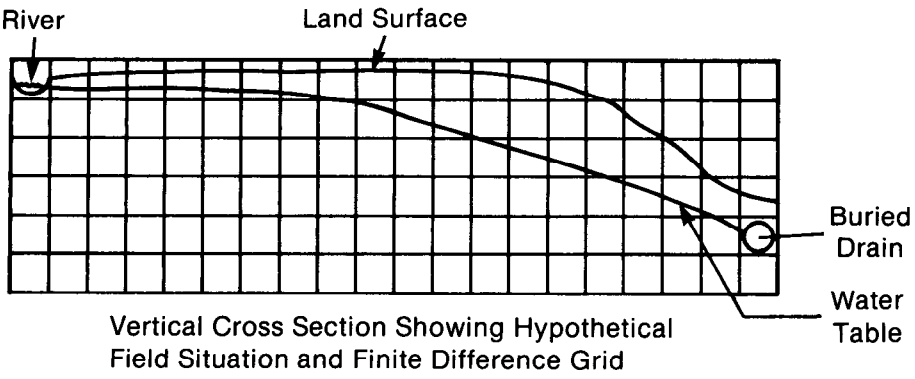
1) if $h_{i,j,k} < (h_{si,j,k} - d_{i,j})$ no changes are made in the terms HCOF or RHS for cell i,j,k;
2) if $h_{i,j,k} > h_{si,j}$, $Q_{ETMi,j}$ is added to $RHS_{i,j}$; and
3) if $(h_{si,j,k} - d_{i,j}) \leq h_{i,j,k} \leq h_{si,j,k}$, $-Q_{ETMi,j}/d_{i,j}$ is added to $HCOF_{i,j,k}$ and $-Q_{ETMi,j} \{\frac{h_{si,j} - d_{i,j}}{d_{i,j}}\}$ is added to RHS i,j,k.

The value of $h_{s,i,j}$, the water table elevation at which evapotranspiration is maximum, should normally be taken as the average land surface elevation in the map area $DELR_j DELC_i$; the cutoff or extinction depth, $d_{i,j}$, is then frequently assumed to be on the order of six to eight feet below land surface (although considerable variation can be introduced by climatic factors, the presence of deep-rooted phreatophytes, or so on). Where the distance from land surface to the water table varies extensively within the area of a cell, care must be exercised in implementing the ET package and in choosing the various parameters of equation (70), or misleading results may be obtained.

The options for selection of the layer from which ET is to be drawn provide some flexibility in adapting the package to special situations, but also require some care in implementation. Figure 43 shows a situation similar to that discussed for the recharge package, in which a cross sectional model has been progressively truncated to follow the water table, using the provision for horizontal conductance formulation under water table conditions (Chapter 5). Figure 43-a shows the hydrologic situation under study, and figure 43-b the final distribution of variable head and inactive (no flow) cells obtained in the simulation.

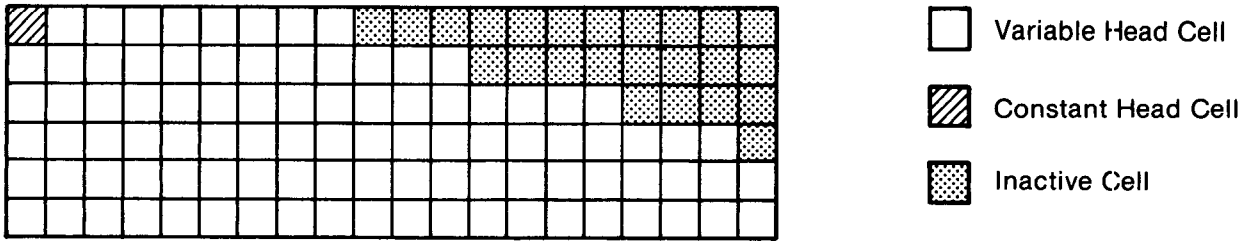
Under option 1 (figure 43-c), evapotranspiration is drawn only from the uppermost layer of the model; in the problem shown, the presence of no flow cells in this layer deletes evapotranspiration from the right half of the model, so that the simulation fails to represent field conditions.

Figure 43-d shows the situation which could be achieved through the use of option 2, assuming that the simulation was carried out in stages and



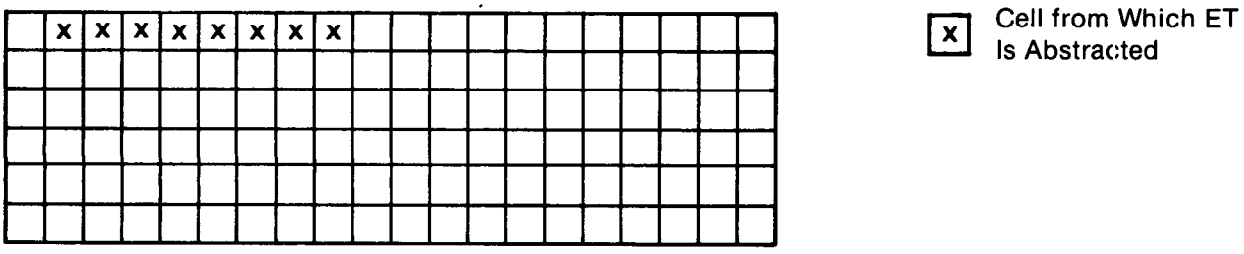
Vertical Cross Section Showing Hypothetical Field Situation and Finite Difference Grid

a



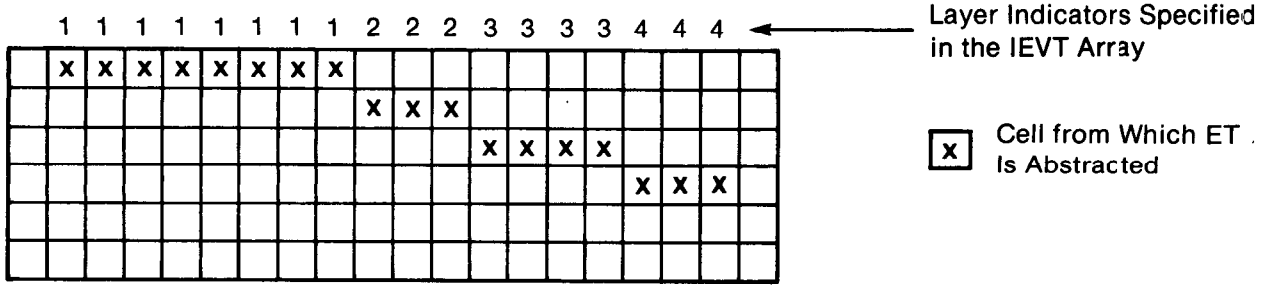
Status of Cells at End of Simulated Period

b



Cells from Which ET Is Abstracted Under Option 1

c



Cells from Which ET Is Abstracted Under Option 2

d

Figure 43.—Hypothetical problem showing cells from which ET will be abstracted under the two options available in the ET Package.

that the user interacted with the simulation process, designating the cells from which evapotranspiration was to be drawn as the truncation of the mesh developed.

Evapotranspiration Package Input

Input to the Evapotranspiration (EVT) Package is read from the unit specified in IUNIT (5).

FOR EACH SIMULATION

EVT1AL

1. Data: NEVTOP IEVTCB
   Format: I10 I10

FOR EACH STRESS PERIOD

EVT1RP

2. Data: INSURF INEVTR INEXDP INIEVT
   Format: I10 I10 I10 I10

3. Data: SURF
   Module: U2DREL

4. Data: EVTR
   Module: U2DREL

5. Data: EXDP
   Module: U2DREL

IF THE ET OPTION IS EQUAL TO TWO

6. Data: IEVT
   Module: U2DINT

Explanation of Fields Used in Input Instructions

NEVTOP--is the evapotranspiration (ET) option code. ET parameters (ET surface, maximum ET rate, and extinction depth) are specified in two-dimensional arrays, SURF, EVTR, and EXDP, with one value for each vertical column. Accordingly, ET is calculated for one cell in each vertical column. The option codes determine for which cell in the column ET will be calculated.

1 - ET is calculated only for cells in the top grid layer.

2 - The cell for each vertical column is specified by the user in array IEVT.

IEVTCB--is a flag and a unit number.

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

If IEVTCB $\leq$ 0, cell-by-cell flow terms will not be printed or recorded.

INSURF--is the ET surface (SURF) read flag.

If INSURF $\geq$ 0, an array containing the ET surface elevation will be read.

If INSURF < 0, the ET surface from the preceding stress period will be reused.

INEVTR--is the maximum ET rate (EVTR) read flag.

If INEVTR $\geq$ 0, an array containing the maximum ET rate will be read.

If INEVTR < 0, the maximum ET rate from the preceding stress period will be reused.

INEXDP--is the extinction depth (EXDP) read flag.

If INEXDP $\geq$ 0, an array containing the extinction depth (EXDP) will be read.

If INEXDP < 0, the extinction depth from the preceding stress period will be reused.

INIEVT--is the layer indicator (IEVT) read flag. It is used only if the ET option (NEVTOP) is equal to two.

If INIEVT $\geq$ 0, an array containing the layer indicators (IEVT) will be read.

If INIEVT < 0, layer indicators used during the preceding stress period will be reused.

SURF--is the elevation of the ET surface.

EVTR--is the maximum ET rate (volume of water per unit area (Lt^{-1})).

EXDP--is the ET extinction depth.

IEVT--is the layer indicator array. For each horizontal location, it indicates the layer from which ET is removed. It is needed only if the ET option is equal to two.

SAMPLE INPUT TO THE EVAPOTRANSPIRATION PACKAGE USING ET OPTION 1

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{NEVTOP,IEVTCB}	1 0
2	Stress period 1--{INSURF, INEVTR, INEXDP, INIEVT}	1 1 1
3	Control record for ET surface array	27 1. (10F5.0) 4
	ET surface	710 715 720 725 730 715 720 725 730 735 720 725 730 735 740 725 730 735 740 745 730 735 740 745 750 735 740 745 750 755
4	Control record for maximum ET rate	0 9.65E-7
5	Control record for extinction depth array	0 10.
2	Stress period 2--{INSURF, INEVTR, INEXDP, INIEVT}	-1 1 -1
4	Control record for maximum ET rate	0 8.23E-7
2	Stress period 3--{INSURF, INEVTR, INEXDP, INIEVT}	-1 1 -1
4	Control record for maximum ET rate	27 9.65E-7 (10F4.0) 4
	max ET rate	1.2 1.2 1.2 1.2 1.3 1.2 1.2 1.2 1.3 1.4 1.2 1.2 1.3 1.4 1.4 1.0 1.0 1.0 1.1 1.1 1.2 1.3 1.3 1.4 1.4 1.3 1.3 1.4 1.4 1.4

SAMPLE INPUT TO THE EVAPOTRANSPIRATION PACKAGE USING ET OPTION 2

DATA ITEM	EXPLANATION	INPUT RECORDS
1	{NEVTOP,IEVTCB}	2 45
2	Stress period 1--{INSURF, INEVTR, INEXDP, INIEVT}	1 1 1 1
3	Control record for ET surface array	27 1. (10F5.0) 4
	ET surface	710 715 720 725 730 715 720 725 730 735 720 725 730 735 740 725 730 735 740 745 730 735 740 745 750 735 740 745 750 755
4	Control record for maximum ET rate	0 9.65E-7
5	Control record for extinction depth array	0 10.
6	Control record for layer indicator array	12 1 (20I2) 3
	Layer numbers	1 2 2 2 3 1 2 2 2 2 1 1 2 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
2	Stress period 2--{INSURF, INEVTR, INEXDP, INIEVT}	-1 1 -1 -1
4	Control record for maximum ET rate	0 8.23E-7

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

Module Documentation for the Evapotranspiration Package

The Evapotranspiration Package (EVT1) consists of four modules, all of which are called by the MAIN program. The modules are:

EVT1AL — Allocates space for arrays to contain maximum ET rate (EVTR), surface elevation (SURF), extinction depth (EXDP), and, if option 2 is specified, the layer indicator (IEVT).

EVT1RP — Reads arrays containing the maximum ET rate (in terms of a volume per unit area), surface elevation, extinction depth, and, if option 2 is specified, the layer indicator. Maximum ET rates are multiplied by cell area to get the maximum ET for each node as a volumetric rate.

EVT1FM — Determines, for each horizontal location, which cell is at the surface. Determines if there is ET from that cell. If there is ET, add the appropriate terms to HCOF and RHS.

EVT1BD — Calculates the rates and accumulated volume of ET out of the flow system.

Narrative for Module EVT1AL

This module allocates space in the X array to store data relating to evapotranspiration.

1. Print a message identifying the package.

2. Read and print the option indicator (NEVTOP) and the unit number for cell-by-cell flow terms (IEVTCB).

3. See if the ET option (NEVTOP) is legal. If NEVTOP is illegal (not 1 or 2), print a message saying the option is illegal. Do not allocate storage. STOP.

4. If NEVTOP is legal, print NEVTOP.

5. If the cell-by-cell flow terms are to be recorded, print the unit number (IEVTCB) where they will be recorded.

6. Allocate space for the maximum ET-rate array (EVTR), the extinction-depth array (EXDP), and the ET-surface array (SURF).

7. If the ET option (NEVTOP) is equal to two, allocate space for a layer-indicator array (IEVT).

8. Calculate and print the number of elements in the X array used by the ET package.

9. RETURN.

Flow Chart for Module EVT1AL

NEVTOP is the ET option.

If NEVTOP = 1, ET is from the top layer.

If NEVTOP = 2, ET is from the layer specified by the user in the indicator array (IEVT).

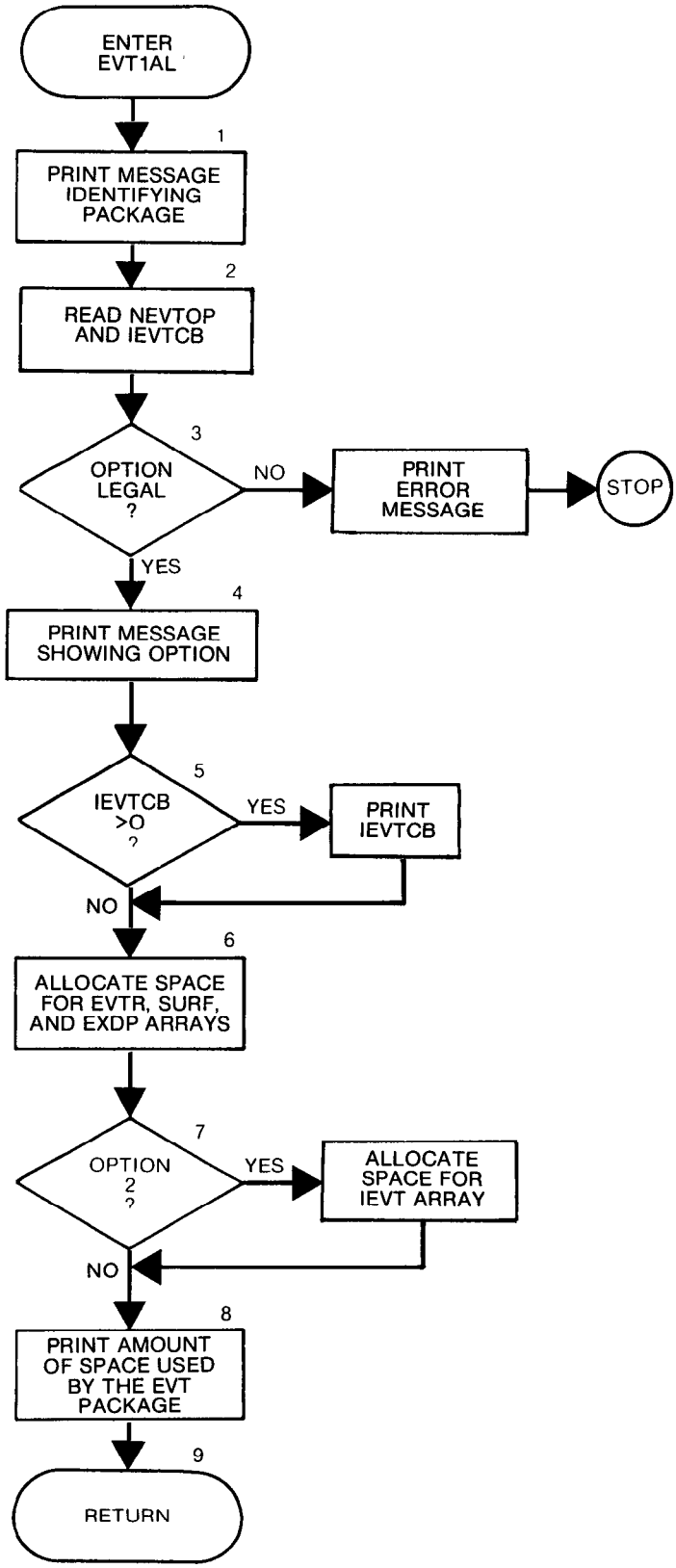
IEVTCB is the unit number on which cell-by-cell flow terms for ET will be written.

EVTR is an array which contains the maximum ET rate for each horizontal cell location.

SURF is an array which contains the elevation of the ET surface.

EXDP is an array which contains the extinction depth for ET.

IEVT is an array which contains the layer number from which ET is taken for each horizontal location. It is used only if option 2 has been specified.



```
      SUBROUTINE EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,LCSURF,
     1                  NCOL,NROW,NEVTOP,IN,IOUT,IEVTCB)
C
C-----VERSION 1607 12MAY1987 EVT1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR EVAPOTRANSPIRATION
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE.
      WRITE(IOUT,1)IN
    1 FORMAT(1HO,'EVT1 -- EVAPOTRANSPIRATION PACKAGE, VERSION 1,',
     1     ' 9/1/87',' INPUT READ FROM UNIT',I3)
C
C2------READ NEVTOP AND IEVTCB.
      READ(IN,3)NEVTOP,IEVTCB
    3 FORMAT(2I10)
C
C3------CHECK TO SEE THAT ET OPTION IS LEGAL.
      IF(NEVTOP.GE.1.AND.NEVTOP.LE.2)GO TO 200
C
C3A-----IF ILLEGAL PRINT A MESSAGE & ABORT SIMULATION.
      WRITE(IOUT,8)
    8 FORMAT(1X,'ILLEGAL ET OPTION CODE. SIMULATION ABORTING')
      STOP
C
C4------IF THE OPTION IS LEGAL THEN PRINT THE OPTION CODE.
  200 IF(NEVTOP.EQ.1) WRITE(IOUT,201)
  201 FORMAT(1X,'OPTION 1 -- EVAPOTRANSPIRATION FROM TOP LAYER')
      IF(NEVTOP.EQ.2) WRITE(IOUT,202)
  202 FORMAT(1X,'OPTION 2 -- EVAPOTRANSPIRATION FROM ONE SPECIFIED',
     1      ' NODE IN EACH VERTICAL COLUMN')
      IRK=ISUM
C
C5------IF CELL-BY-CELL TERMS TO BE SAVED THEN PRINT UNIT NUMBER.
      IF(IEVTCB.GT.0) WRITE(IOUT,203) IEVTCB
  203 FORMAT(1X,'CELL-BY-CELL FLOW TERMS WILL BE SAVED ON UNIT',I3)
C
C6------ALLOCATE SPACE FOR THE ARRAYS EVTR, EXDP AND SURF.
      LCEVTR=ISUM
      ISUM=ISUM+NCOL*NROW
      LCEXDP=ISUM
      ISUM=ISUM+NCOL*NROW
      LCSURF=ISUM
      ISUM=ISUM+NCOL*NROW
C
C7------IF OPTION 2 THEN ALLOCATE SPACE FOR THE INDICATOR ARRAY(IEVT)
      LCIEVT=ISUM
      IF(NEVTOP.NE.2)GO TO 300
      ISUM=ISUM+NCOL*NROW
C
C8------CALCULATE & PRINT AMOUNT OF SPACE USED BY ET PACKAGE.
  300 IRK=ISUM-IRK
      WRITE(IOUT,4)IRK
    4 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED FOR EVAPOTRANSPIRATION')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX)WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE MADE LARGER***')
C
C9------RETURN.
      RETURN
      END
```

List of Variables for Module EVT1AL

Variable	Range	Definition
IEVTCB	Package	Flag. If IEVTCB > 0 and ICBCFL ≠ 0, cell-by-cell flow terms for the EVT1 Package will be recorded on UNIT = IEVTCB.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IRK	Module	Before this module allocates space, IRK is set equal to ISUM. After allocation, IRK is subtracted from ISUM to get the amount of space in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	Index number of the last element of the X array allocated by this module.
LCEVTR	Package	Location in the X array of the first element of array EVTR.
LCEXDP	Package	Location in the X array of the first element of array EXDP.
LCIEVT	Package	Location in the X array of the first element of array IEVT.
LCSURF	Package	Location in the X array of the first element of array SURF.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option: = 1, ET is from the top layer. = 2, ET at each horizontal-cell location is from the layer specified by the user in the layer-indicator array (IEVT).
NROW	Global	Number of rows in the grid.

Narrative for Module EVT1RP

This module reads data used to calculate the terms which represent evapotranspiration.

1. Read the values INSURF, INEXDP, INEVTR, and INIEVT which indicate whether the data contained in arrays SURF, EXDP, EVTR, and IEVT, respectively, used during the last stress period, are to be used for the current stress period.

2. Test INSURF to see where the ET-surface array (SURF) is coming from. If INSURF is less than zero, the ET-surface elevation used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 4.

3. INSURF is greater than or equal to zero. CALL U2DREL to read SURF.

4. Test INEVTR to see where the maximum ET rate (EVTR) is coming from. If INEVTR is less than zero, the maximum ET rate used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 7.

5. INEVTR is greater than or equal to zero. CALL U2DREL to read the maximum ET rate (EVTR).

6. Multiply the maximum ET rate by the area to get a volumetric rate.

7. Test INEXDP to see where the extinction rate is coming from. If INEXDP is less than zero, the extinction depth used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 9.

8. If INEXDP is greater than or equal to zero, CALL U2DREL to read the extinction depth.

9. If the ET option (NEVTOP) is equal to two, a layer-indicator array is needed.

10. Test INIEVT to see where the layer indicator is coming from. If INIEVT is less than zero, the indicator array used in the last stress period will be used again in this stress period. Print a message to that effect and GO TO 12.

11. If INIEVT is greater than or equal to zero, CALL U2DINT to read the IEVT array.

12. RETURN.

Flow Chart for Module EVT1RP

INEVTR is a flag which, when set, indicates that the maximum ET rate EVTR should be read for the current stress period. If it is clear (less than zero), maximum ET rates from the last stress period should be reused.

INIEVT, INSURF, and INEXDP are flags similar to INEVTR used for the layer indicator array (IEVT), the ET surface array (SURF), and the extinction depth array (EXDP), respectively.

EVTR is an array containing the maximum ET rate for every horizontal cell location.

SURF is an array containing the ET surface elevation for each horizontal cell location.

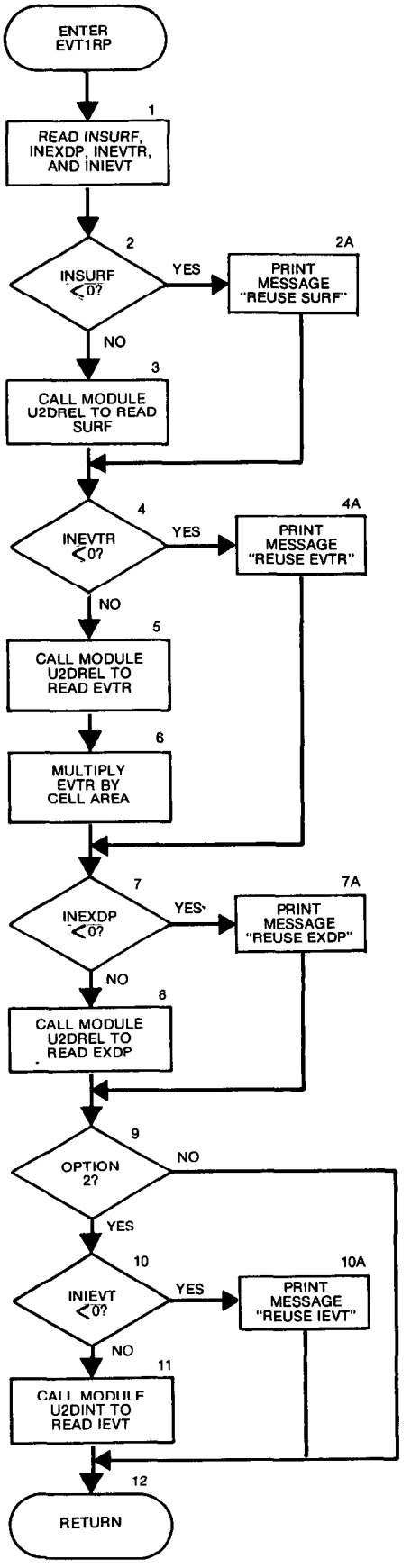
EXDP is an array containing the extinction depth for each horizontal cell location.

IEVT is an array containing a layer indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location from which ET is taken. It is used only if the ET option (NEVTOP) is equal to two.

NEVTOP is the ET option.

If NEVTOP = 1, ET is from the top layer.

If NEVTOP = 2, ET is from the layer specified by the user in the indicator array (IEVT).



```
      SUBROUTINE EVT1RP(NEVTOP,IEVT,EVTR,EXDP,SURF,DELR,DELC,
     1                  NCOL,NROW,IN,IOUT)
C
C-----VERSION 1635 24JUL1987 EVT1RP
C     ******************************************************************
C     READ EVAPOTRANSPIRATION DATA
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 ANAME
      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
     1          SURF(NCOL,NROW),ANAME(6,4),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) /'    ',' ',' ',' ET',' LAY','ER I','NDEX'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1  ANAME(6,2) /'    ',' ',' ',' ET',' SUR','FACE'/
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
     1  ANAME(6,3) /' EVA','POTR','ANSP','IRAT','ION ','RATE'/
      DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),
     1  ANAME(6,4) /'    ',' ','EXTI','NCTI','ON D','EPTH'/
C     ------------------------------------------------------------------
C
C1------READ FLAGS SHOWING WHETHER DATA IS TO BE REUSED.
      READ(IN,6)INSURF,INEVTR,INEXDP,INIEVT
    6 FORMAT(4I10)
C
C2------TEST INSURF TO SEE WHERE SURFACE ELEVATION COMES FROM.
      IF(INSURF.GE.0)GO TO 32
C
C2A-----IF INSURF<0 THEN REUSE SURFACE ARRAY FROM LAST STRESS PERIOD
      WRITE(IOUT,3)
    3 FORMAT(1HO,'REUSING SURF FROM LAST STRESS PERIOD')
      GO TO 35
C
C3------IF INSURF=>0 THEN CALL MODULE U2DREL TO READ SURFACE.
   32 CALL U2DREL(SURF,ANAME(1,2),NROW,NCOL,0,IN,IOUT)
C
C4------TEST INEVTR TO SEE WHERE MAX ET RATE COMES FROM.
   35 IF(INEVTR.GE.0)GO TO 37
C
C4A-----IF INEVTR<0 THEN REUSE MAX ET RATE.
      WRITE(IOUT,4)
    4 FORMAT(1HO,'REUSING EVTR FROM LAST STRESS PERIOD')
      GO TO 45
C
C5------IF INEVTR=>0 CALL MODULE U2DREL TO READ MAX ET RATE.
   37 CALL U2DREL(EVTR,ANAME(1,3),NROW,NCOL,0,IN,IOUT)
C
C6------MULTIPLY MAX ET RATE BY CELL AREA TO GET VOLUMETRIC RATE
      DO 40 IR=1,NROW
      DO 40 IC=1,NCOL
      EVTR(IC,IR)=EVTR(IC,IR)*DELR(IC)*DELC(IR)
   40 CONTINUE
C
C7------TEST INEXDP TO SEE WHERE EXTINCTION DEPTH COMES FROM
   45 IF(INEXDP.GE.0)GO TO 47
C
C7A-----IF INEXDP<0 REUSE EXTINCTION DEPTH FROM LAST STRESS PERIOD
      WRITE(IOUT,5)
    5 FORMAT(1HO,'REUSING EXDP FROM LAST STRESS PERIOD')
      GO TO 48
C
C8------IF INEXDP=>0 CALL MODULE U2DREL TO READ EXTINCTION DEPTH
   47 CALL U2DREL(EXDP,ANAME(1,4),NROW,NCOL,0,IN,IOUT)
C
C9------IF OPTION(NEVTOP) IS 2 THEN WE NEED AN INDICATOR ARRAY.
  48  IF(NEVTOP.NE.2)GO TO 50
C
C10-----IF INIEVT<0 THEN REUSE LAYER INDICATOR ARRAY.
      IF(INIEVT.GE.0)GO TO 49
      WRITE(IOUT,2)
    2 FORMAT(1HO,'REUSING IEVT FROM LAST STRESS PERIOD')
      GO TO 50
C
C11-----IF INIEVT=>0 THEN CALL MODULE U2DINT TO READ INDICATOR ARRAY.
   49 CALL U2DINT(IEVT,ANAME(1,1),NROW,NCOL,0,IN,IOUT)
C
C12-----RETURN
   50 RETURN
      END
```

List of Variables for Module EVT1RP

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.
EVTR	Package	DIMENSION (NCOL,NROW), Maximum ET rate.
EXDP	Package	DIMENSION (NCOL,NROW), Extinction depth.
IC	Module	Index for columns.
IEVT	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal cell location from which ET will be taken if the ET option (NEVTOP) is equal to two.
IN	Package	Primary unit number from which input for this package will be read.
INEVTR	Module	Flag. $\geq$ 0, EVTR array will be read. < 0, EVTR array already in memory from the last stress period will be used.
INEXDP	Module	Flag. $\geq$ 0, EXDP array will be read. < 0, EXDP array already in memory from the last stress period will be used.
INIEVT	Module	Flag. $\geq$ 0, IEVT array will be read. < 0, IEVT array already in memory from the last stress period will be used.
INSURF	Module	Flag. $\geq$ 0, SURF array will be read. < 0, SURF array already in memory from the last stress period will be used.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option. = 1, ET is from the top layer. = 2, ET at each horizontal-cell location is from the layer specified in the layer-indicator array (IEVT).
NROW	Global	Number of rows in the grid.
SURF	Package	DIMENSION (NCOL,NROW), Elevation of the ET surface.

Narrative for Module EVT1FM

This module adds terms representing ET to the finite-difference equations.

1. For each horizontal-cell location, determine which layer ET comes from and add the appropriate terms to the equation for the cell. DO STEPS 1-7.

2. Set the layer index equal to one.

3. If option 2 was invoked, get the layer index from the indicator array (IEVT).

4. If the cell is external, move on to the next horizontal-cell location. SKIP STEPS 5-7.

5. If the head in the aquifer is greater than or equal to the ET-surface elevation, add EVTR to RHS and move on to the next horizontal-cell location. SKIP STEPS 6 AND 7.

6. If the head in the aquifer is less than the extinction elevation (ET surface minus extinction depth), no terms need to be added to the finite-difference equation. Move on to the next horizontal-cell location. SKIP STEP 7.

7. Add the term -EVTR/EXDP to HCOF and subtract the term -EVTR(EXDP - SURF)/EXDP from RHS.

8. RETURN.

Flow Chart for Module EVT1FM

IEVT is an array containing a layer indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location from which ET is taken. It is used only if the ET option (NEVTOP) is equal to two.

SURF is an array containing the maximum ET rate for every horizontal cell location.

EVTR is an array containing the maximum ET rate for every horizontal cell location.

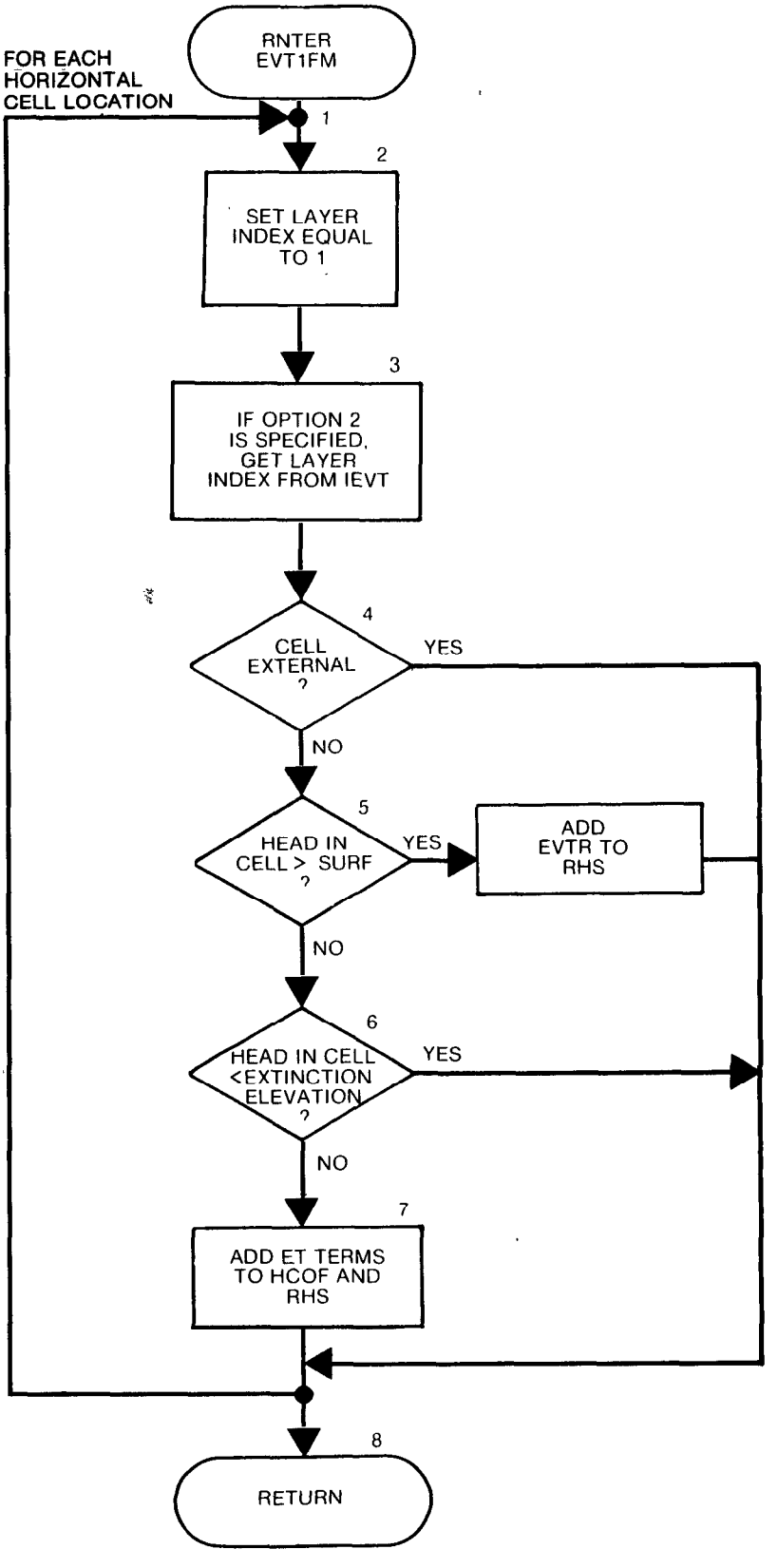
RHS is an accumulator in which the right hand side of the finite-difference equation is formulated.

HCOF is an accumulator in which a coefficient of head in the finite-difference equation is formulated.

NEVTOP is the ET option.

If NEVTOP = 1, ET is from the top layer.

If NEVTOP = 2, ET is from the layer specified by the user in the indicator array (IEVT).



```
      SUBROUTINE EVT1FM(NEVTOP,IEVT,EVTR,EXDP,SURF,RHS,HCOF,
     1                  IBOUND,HNEW,NCOL,NROW,NLAY)
C
C-----VERSION 1031 10APR1985 EVT1FM
C     ******************************************************************
C        ADD EVAPOTRANSPIRATION TO RHS AND HCOF
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
     1          SURF(NCOL,NROW),RHS(NCOL,NROW,NLAY),
     2          HCOF(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     3          HNEW(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------PROCESS EACH HORIZONTAL CELL LOCATION
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C2------SET THE LAYER INDEX EQUAL TO 1
      IL=1
C
C3------IF OPTION 2 IS SPECIFIED THEN GET LAYER INDEX FROM IEVT ARRAY
      IF(NEVTOP.EQ.2)IL=IEVT(IC,IR)
C
C4------IF THE CELL IS EXTERNAL IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 10
      C=EVTR(IC,IR)
      S=SURF(IC,IR)
      H=HNEW(IC,IR,IL)
C
C5------IF AQUIFER HEAD IS GREATER THAN OR EQUAL TO SURF, ET IS CONSTANT
      IF(H.LT.S) GO TO 5
C
C5A-----SUBTRACT -EVTR FROM RHS
      RHS(IC,IR,IL)=RHS(IC,IR,IL) + C
      GO TO 10
C
C6------IF DEPTH TO WATER>=EXTINCTION DEPTH THEN ET IS 0
    5 D=S-H
      X=EXDP(IC,IR)
      IF(D.GE.X)GO TO 10
C
C7------LINEAR RANGE. ADD ET TERMS TO BOTH RHS AND HCOF.
      RHS(IC,IR,IL)=RHS(IC,IR,IL)+C-C*S/X
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C/X
   10 CONTINUE
C
C8------RETURN
      RETURN
      END
```

List of Variables for Module EVT1FM

Variable	Range	Definition
C	Module	Maximum ET rate.
D	Module	Depth to water.
EVTR	Package	DIMENSION (NCOL,NROW), Maximum ET rate.
EXDP	Package	DIMENSION (NCOL,NROW), Extinction depth.
H	Module	Head in the cell.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IEVT	Package	DIMENSION (NCOL,NROW), Layer number, for each horizontal-cell location, from which ET will be taken if the ET option (NEVTOP) is equal to two.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option. = 1, ET is from the top layer. = 2, ET at each horizontal cell location is from the layer specified in the layer-indicator array (IEVT).
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of finite-difference equation. RHS is an accumulation of terms from several different packages.
S	Module	ET surface elevation for a cell.
SURF	Package	DIMENSION (NCOL,NROW), Elevation of the ET surface.
X	Module	Extinction depth for a cell.

Narrative for Module EVT1BD

This module calculates rates and volumes removed from the aquifer by evapotranspiration.

1. Clear the rate accumulator RATOUT.

2. If budget terms will be saved, clear the buffer (BUFF) in which they will be accumulated.

3. Process each horizontal-cell location one at a time calculating flow to evapotranspiration (STEPS 4-11).

4. Set the layer index (IL) equal to one.

5. If option 2 is in effect, get the layer index from the layer-indicator array (IEVT).

6. If the cell is external (IBOUND $\leq$ 0), bypass processing of the cell.

7. If the head in the aquifer is greater than the elevation of the ET surface, set the ET rate for the cell equal to the maximum ET rate. SKIP STEPS 8 AND 9.

8. If the depth to the water is greater than the extinction depth, bypass further processing of this cell. SKIP STEP 9.

9. Calculate the ET flow into the model using the linear approximation.

10. Subtract the ET flow from the accumulator (RATOUT).

11. If the cell-by-cell flow terms are to be saved, add the ET rate to the buffer (BUFF).

12. If the cell-by-cell flow terms are to be saved, call module UBUDSV to write the buffer (BUFF) onto a disk.

13. Move RATOUT into the VBVL array for printing by BAS1OT.

14. Add RATOUT multiplied by the time-step length to the volume accumulators in VBVL for printing by BAS1OT.

15. Move the ET budget-term labels to VBNM for printing by BAS1OT.

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

17. RETURN.

Flow Chart for Module EVT1BD

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

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

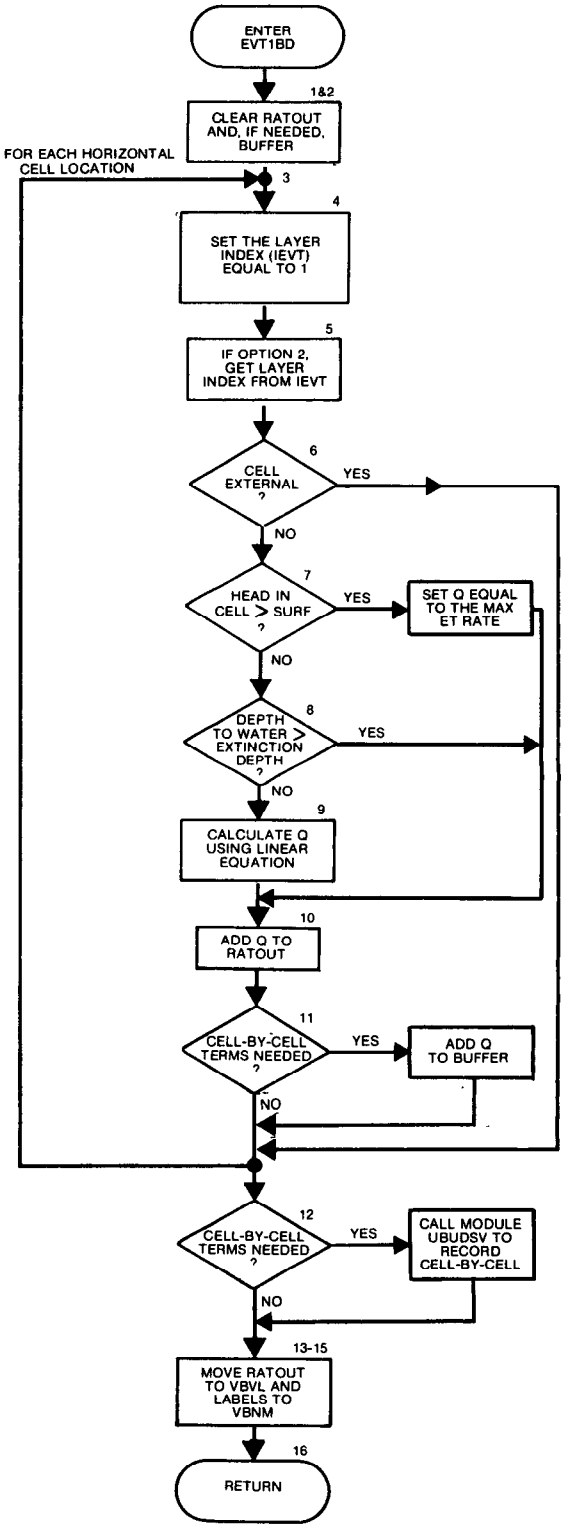
IEVT is an array containing a layer indicator for each horizontal cell location. For each horizontal cell location, it indicates the layer number of the cell at that location from which ET is taken. It is used only if NEVTOP is equal to two.

SURF is an array containing the ET surface elevation for each horizontal cell location.

Q is the flow to ET from an individual cell.

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

VBNM is a table of labels for budget terms.



```
      SUBROUTINE EVT1BD(NEVTOP,IEVT,EVTR,EXDP,SURF,IBOUND,HNEW,
     1            NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,KSTP,KPER,
     2            IEVTCB,ICBCFL,BUFF,IOUT)
C-----VERSION 1608 12MAY1987 EVT1BD
C     ******************************************************************
C     CALCULATE VOLUMETRIC BUDGET FOR EVAPOTRANSPIRATION
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      DOUBLE PRECISION HNEW
      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
     1          SURF(NCOL,NROW),IBOUND(NCOL,NROW,NLAY),
     2          VBVL(4,20),VBNM(4,20),HNEW(NCOL,NROW,NLAY),
     3          BUFF(NCOL,NROW,NLAY)
      DIMENSION TEXT(4)
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','    ','  ET'/
C     ------------------------------------------------------------------
C
C1------CLEAR THE RATE ACCUMULATOR.
      RATOUT=0
C
C2------IF CELL-BY-CELL FLOW TERMS WILL BE SAVED THEN CLEAR THE BUFFER.
      IBD=0
      IF(IEVTCB.LE.0 .OR. ICBCFL.EQ.0) GO TO 5
      IBD=1
      DO 4 IL=1,NLAY
      DO 4 IR=1,NROW
      DO 4 IC=1,NCOL
      BUFF(IC,IR,IL)=0.
    4 CONTINUE
C
C3------PROCESS EACH HORIZONTAL CELL LOCATION
    5 DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
C4------SET THE LAYER INDEX EQUAL TO 1
      IL=1
C
C5------IF OPTION 2 IS SPECIFIED THEN GET LAYER INDEX FROM IEVT ARRAY
      IF(NEVTOP.EQ.2)IL=IEVT(IC,IR)
C
C6------IF CELL IS EXTERNAL THEN IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 10
      C=EVTR(IC,IR)
      S=SURF(IC,IR)
```

```
      H=HNEW(IC,IR,IL)
C
C7------IF AQUIFER HEAD => SURF,SET Q=MAX ET RATE
      IF(H.LT.S) GO TO 7
      Q=-C
      GO TO 9
C
C8------IF DEPTH=>EXTINCTION DEPTH, ET IS 0
    7 X=EXDP(IC,IR)
      D=S-H
      IF(D.GE.X)GO TO 10
C
C9------LINEAR RANGE . Q=-EVTR(H-EXEL)/EXDP
      Q=C*D/X-C
C
C10-----ACCUMULATE TOTAL FLOW RATE
    9 RATOUT=RATOUT-Q
C
C11-----IF CELL-BY-CELL FLOW TERMS TO BE SAVED THE ADD Q TO BUFFER.
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=Q
   10 CONTINUE
C
C12-----IF C-B-C TO BE SAVED CALL MODULE UBUDSV TO RECORD THEM.
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IEVTCB,BUFF,NCOL,NROW,
     1                         NLAY,IOUT)
C
C13-----MOVE TOTAL ET RATE INTO VBVL FOR PRINTING BY BAS1OT.
      VBVL(3,MSUM)=0.
      VBVL(4,MSUM)=RATOUT
C
C14-----ADD ET(ET_RATE TIMES STEP LENGTH) TO VBVL
      VBVL(1,MSUM)=0.
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
C
C15-----MOVE BUDGET TERM LABELS TO VBNM FOR PRINT BY MODULE BAS1OT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C16-----INCREMENT BUDGET TERM COUNTER
      MSUM=MSUM+1
C
C17-----RETURN
      RETURN
      END
```

List of Variables for Module EVT1BD

Variable	Range	Definition
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
C	Module	Maximum ET rate at a cell.
D	Module	Depth to water below the ET surface.
DELT	Global	Length of the current time step.
EVTR	Package	DIMENSION (NCOL,NROW), Maximum ET rate.
EXDP	Package	DIMENSION (NCOL,NROW), Extinction depth.
H	Module	Head in the cell.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBD	Module	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be recorded for the current time step.
IEVT	Package	DIMENSION (NCOL,NROW), Layer number for each horizontal-cell location from which ET will be taken if the ET option (NEVTOP) is equal to two.
IEVTCB	Package	Flag. If IEVTCB > 0 and ICBCFL ≠ 0, cell-by-cell flow terms for the EVT1 Package will be recorded on UNIT = IEVTCB.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.
NEVTOP	Package	ET option. = 1, ET is from the top layer. = 2, ET at each horizontal-cell location is from the layer specified in the layer-indicator array (IEVT).
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
Q	Module	Flow from ET into the cell. (Reverse the sign to get the flow to ET.)

List of Variables for Module EVT1BD (Continued)

Variable	Range	Definition
RATOUT	Module	Accumulator for the total flow out of the flow field to ET.
S	Module	Elevation of the ET surface for a cell.
SURF	Package	DIMENSION (NCOL,NROW), Elevation of the ET surface.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the current time step out of the flow field. (3,N), Volume into the flow field during simulation. (4,N), Volume out of the flow field during simulation.
X	Module	Extinction depth for a cell.

CHAPTER 11

GENERAL-HEAD BOUNDARY PACKAGE

Conceptualization and Implementation

The function of the General-Head Boundary (GHB) Package is mathematically similar to that of the River, Drain and ET Packages, in that flow into or out of a cell i,j,k, from an external source is provided in proportion to the difference between the head in the cell, $h_{i,j,k}$, and the head assigned to the external source, $h_{bi,j,k}$. Thus a linear relationship between flow into the cell and head in the cell is established, i.e.

$$Q_{bi,j,k} = C_{bi,j,k} (h_{bi,j,k} - h_{i,j,k}) \qquad (78)$$

where $Q_{bi,j,k}$ is the flow into cell i,j,k from the source; $C_{bi,j,k}$ is the conductance between the external source and cell i,j,k; $h_{bi,j,k}$ is the head assigned to the external source; and $h_{i,j,k}$ is the head in cell i,j,k. The relationship between cell i,j,k and the external source is shown schematically in figure 44. The constant-head source is represented by the apparatus on the right in figure 44, which holds the source head at the level h_b regardless of other factors; the link between the source and cell i,j,k is represented by the block of porous material $C_{bi,j,k}$. Note that figure 44 shows no mechanism to limit flow in either direction as $h_{i,j,k}$ rises or falls.

A graph of $Q_{bi,j,k}$ versus $h_{i,j,k}$ as given by equation (78) is shown in figure 45. In contrast to the River, Drain and ET Packages, the GHB Package provides no limiting value of flow to bound the linear function in either direction; and as the head difference between cell i,j,k and the source increases, flow into or out of the cell continues to increase without

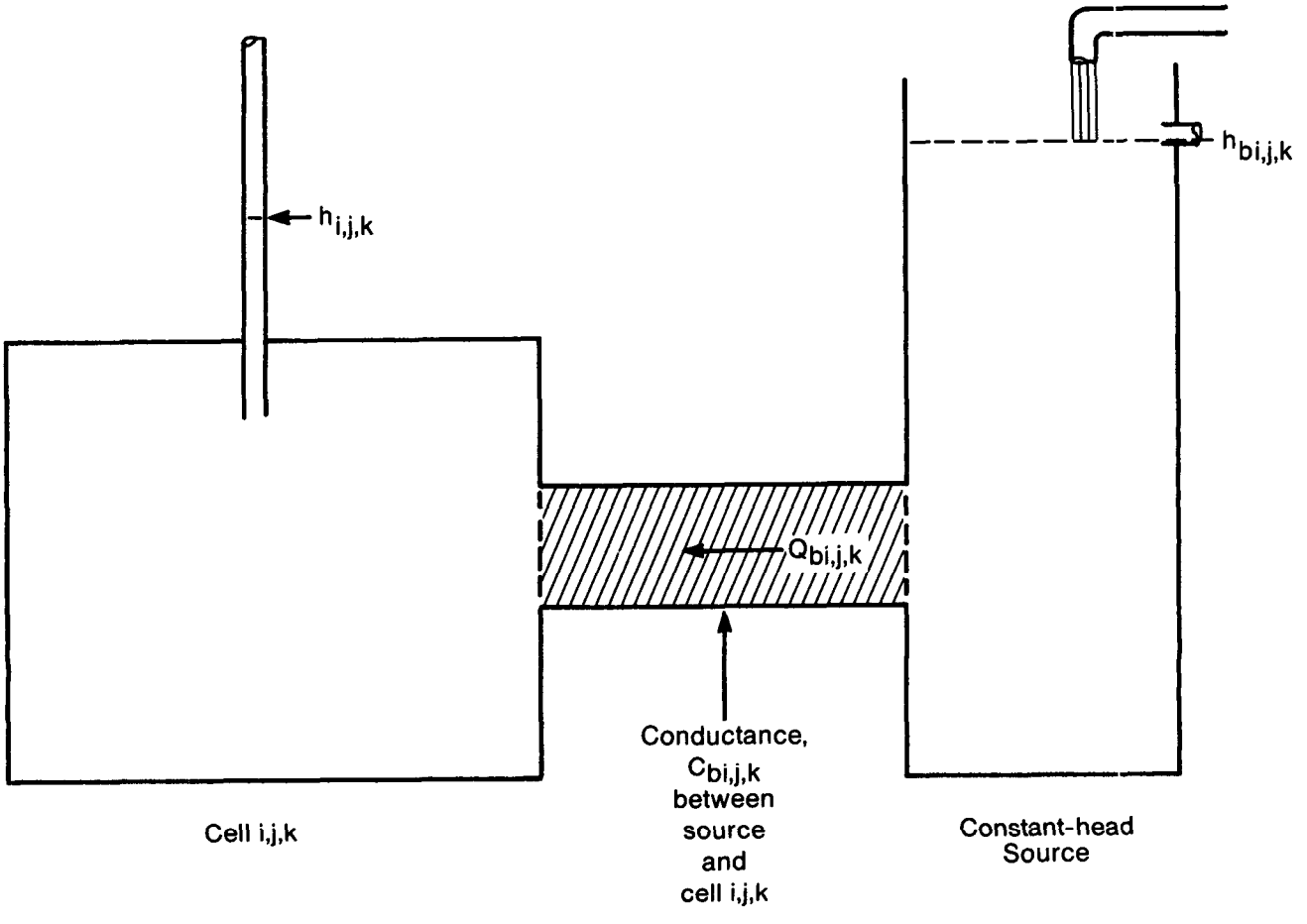


Figure 44.—Schematic diagram illustrating principle of general-head boundary package.

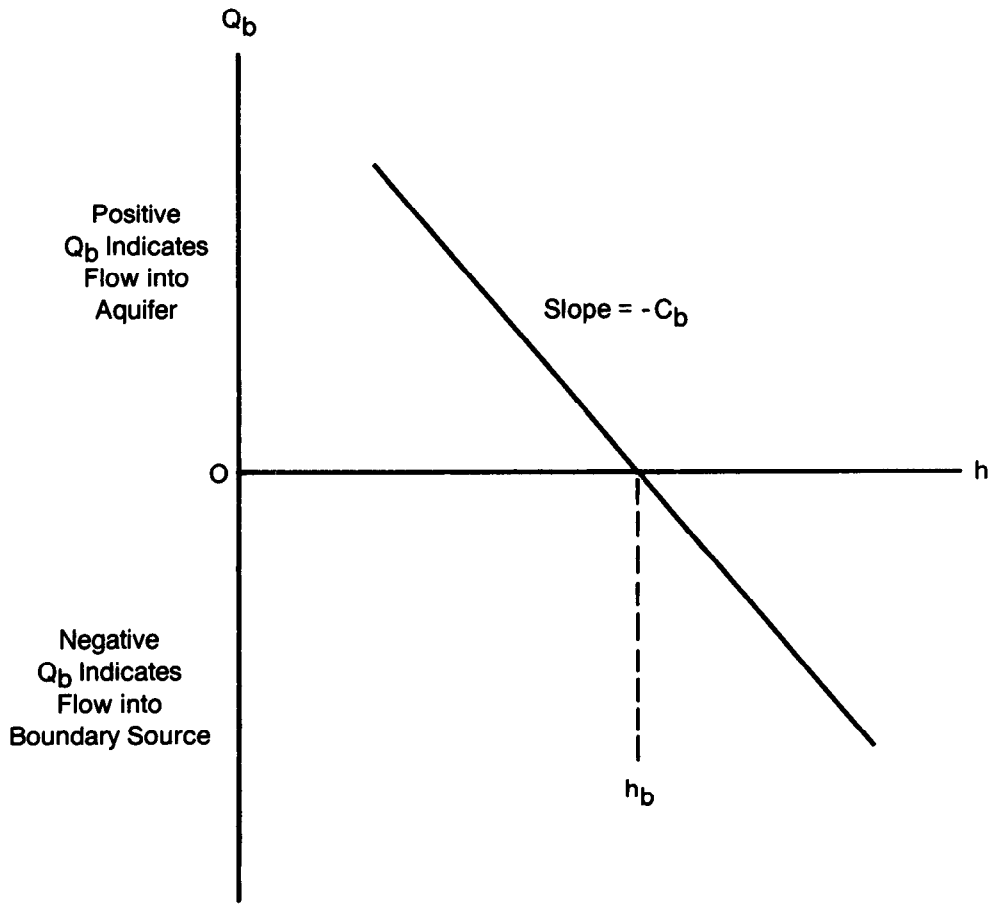


Figure 45.—Plot of flow, Q_b, from a general-head boundary source into a cell as a function of head, h, in the cell where h_b is the source head.

limit. Care must accordingly be used in utilizing the GHB Package to insure that unrealistic flows into or out of the system do not develop during the course of simulation.

Because $Q_{bi,j,k}$ of equation (78) is defined as an inflow to the aquifer it must be added to the left side of equation (24). In terms of the expressions HCOF and RHS of equation (26), this is accomplished in the model by subtracting the term $C_{bi,j,k}$ from $HCOF_{i,j,k}$ and subtracting the term $C_{bi,j,k}h_{bi,j,k}$ from $RHS_{i,j,k}$ as the matrix equations are assembled.

General-Head Boundary Package Input

Input for the General-Head Boundary (GHB) Package is read from the unit specified in IUNIT(7).

FOR EACH SIMULATION

GHB1AL

1. Data: MXBND IGHBCB
   Format: I10 I10

FOR EACH STRESS PERIOD

GHB1RP

2. Data: ITMP
   Format: I10

3.	Data:	Layer	Row	Column	Boundary Head	Cond
	Format:	I10	I10	I10	F10.0	F10.0

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

Explanation of Fields Used in Input Instructions

MXBND--is the maximum number of general-head boundary cells at one time.

IGHBCB--is a flag and a unit number.

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

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

If IGHBCB < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.

If ITMP < 0, GHB data from the preceding stress period will be reused.

If ITMP $\geq$ 0, ITMP is the number of general-head boundaries during the current stress period.

Layer--is the layer number of the cell affected by the head-dependent boundary.

Row--is the row number of the cell affected by the head-dependent boundary.

Column--is the column number of the cell affected by the head-dependent boundary.

Boundary head--is the head on the boundary.

Cond--is the hydraulic conductance of the interface between the aquifer cell and the boundary.

SAMPLE INPUT TO THE GENERAL HEAD BOUNDARY PACKAGE

DATA ITEM	EXPLANATION	INPUT RECORDS				
1	{MXBND, IGHBCB}_____	6	24			
2	STRESS PERIOD 1 {ITMP}_____	4				
3	FIRST BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	5	6	235.0	.0012
3	SECOND BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	4	6	230.0	.0012
3	THIRD BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	5	8	250.0	.0018
3	FOURTH BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	7	6	235.0	.0012
2	STRESS PERIOD 2 {ITMP}_____	-1				
2	STRESS PERIOD 3 {ITMP}_____	-1				
2	STRESS PERIOD 4 {ITMP}_____	6				
3	FIRST BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	5	6	235.0	.0012
3	SECOND BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	4	6	230.0	.0012
3	THIRD BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	5	8	250.0	.0018
3	FOURTH BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	7	6	235.0	.0018
3	FIFTH BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	9	6	235.0	.0012
3	SIXTH BOUNDARY {Layer, Row, Column, Head, Conductance_____	2	10	6	250.0	.0012

Module Documentation for the General-Head Boundary Package

The General-Head Boundary Package (GHB1) consists of four modules, all of which are called by the MAIN program. The modules are:

GHB1AL — Allocates space for an array that contains the general-head boundary list (BNDS).

GHB1RP — Reads location, boundary head, and boundary conductance (C_m) of each cell containing general-head boundary m.

GHB1FM — Adds the terms $-C_m$ and $-C_mHB_m$ to the accumulators $HCOF_{i,j,k}$ and $RHS_{i,j,k}$, respectively.

GHB1BD — Calculates the rates and accumulated volume of flow to and from general-head boundaries.

Narrative for Module GHB1AL

This module allocates space in the X array to store the list of general-head boundaries (GHB).

1. Print a message identifying the package and initialize NBOUND (number of general-head boundaries).

2. Read and print MXBND (the maximum number of general-head boundaries) and IGHBCB (the unit number for saving cell-by-cell flow terms or a flag indicating that cell-by-cell flow terms should be printed).

3. Set LCBNDS, which will point to the first element in the boundary list (BNDS), equal to ISUM which is currently pointing to the first unallocated element in the X array.

4. Calculate the amount of space needed for the boundary list (five values for each boundary--row, column, layer, head, and conductance) and add it to ISUM so that it continues to point to the first unallocated element in X.

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

6. RETURN.

Flow Chart for Module GHB1AL

NBOUND is the number of general-head boundaries being simulated at any given time.

MXBND is the maximum number of general-head boundaries simulated.

IGHBCB is a flag and a unit number.

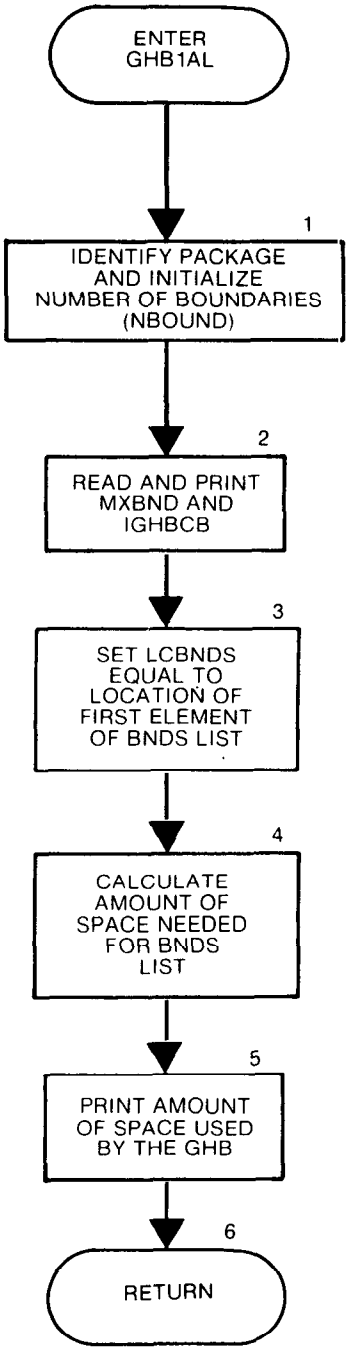
If IGHBCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

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

If IGHBCB < 0, the boundary leakage for each cell will be printed whenever ICBCFL is set.

LCBNDS is the location in the X array of the list of general-head boundaries data (BNDS).

BNDS is a table containing data for general-head boundaries.



```
      SUBROUTINE GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,IN,IOUT,
     1                  IGHBCB)
C
C-----VERSION 1610 12MAY1987 GHB1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR HEAD-DEPENDENT BOUNDARIES
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE # OF GENERAL HEAD BOUNDS
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'GHB1 -- GHB PACKAGE, VERSION 1, 9/1/87',
     1' INPUT READ FROM UNIT',I3)
      NBOUND=0
C
C2------READ AND PRINT MXBND AND IGHBCB (MAX # OF BOUNDS AND UNIT
C2------FOR CELL-BY-CELL FLOW TERMS FOR GHB)
      READ(IN,2) MXBND,IGHBCB
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXBND
    3 FORMAT(1H ,'MAXIMUM OF',I5,' HEAD-DEPENDENT BOUNDARY NODES')
      IF(IGHBCB.GT.0) WRITE(IOUT,9) IGHBCB
    9 FORMAT(1X,'CELL-BY-CELL FLOW WILL BE RECORDED ON UNIT',I3)
      IF(IGHBCB.LT.0) WRITE(IOUT,8)
    8 FORMAT(1X,'CELL-BY-CELL FLOW WILL BE PRINTED WHEN ICBCFL NOT 0')
C
C3------SET LCBNDS EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCBNDS=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY THE GENERAL HEAD LIST.
      ISP=5*MXBND
      ISUM=ISUM+ISP
C
C5------PRINT AMOUNT OF SPACE USED BY THE GHB PACKAGE
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED FOR HEAD',
     1      '-DEPENDENT BOUNDARIES')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6------RETURN
      RETURN
      END
```

List of Variables for Module GHB1AL

Variable	Range	Definition
IGHBCB	Package	Flag and a unit number. > 0, unit number on which the cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, boundary leakage for each cell will be printed whenever IGHBFL is set.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISP	Module	Number of words in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM-1.
LCBNDS	Package	Location in the X array of the first element of array BNDS.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current stress period.

Narrative for Module GHB1RP

This module reads data to build the general-head boundary list.

1. Read ITMP. ITMP is the number of general-head boundaries or a flag indicating that data from the previous stress period should be reused.

2. Test ITMP. If ITMP is less than zero, the general-head boundary data read for the last stress period will be reused. Print a message to that effect and RETURN.

3. If ITMP is greater than or equal to zero, it is the number of general-head boundaries for this stress period. Set the number of general-head boundaries (NBOUND) in the current stress period equal to ITMP.

4. Compare the number of general-head boundaries (NBOUND) in the current stress period to the number specified as the maximum for the simulation (MXBND). If NBOUND is greater than MXBND, STOP.

5. Print the number of general-head boundaries in the current stress period (NBOUND).

6. See if there are any general-head boundaries. If there are none in the current stress period (NBOUND = 0), bypass further boundary processing (SKIP STEP 7).

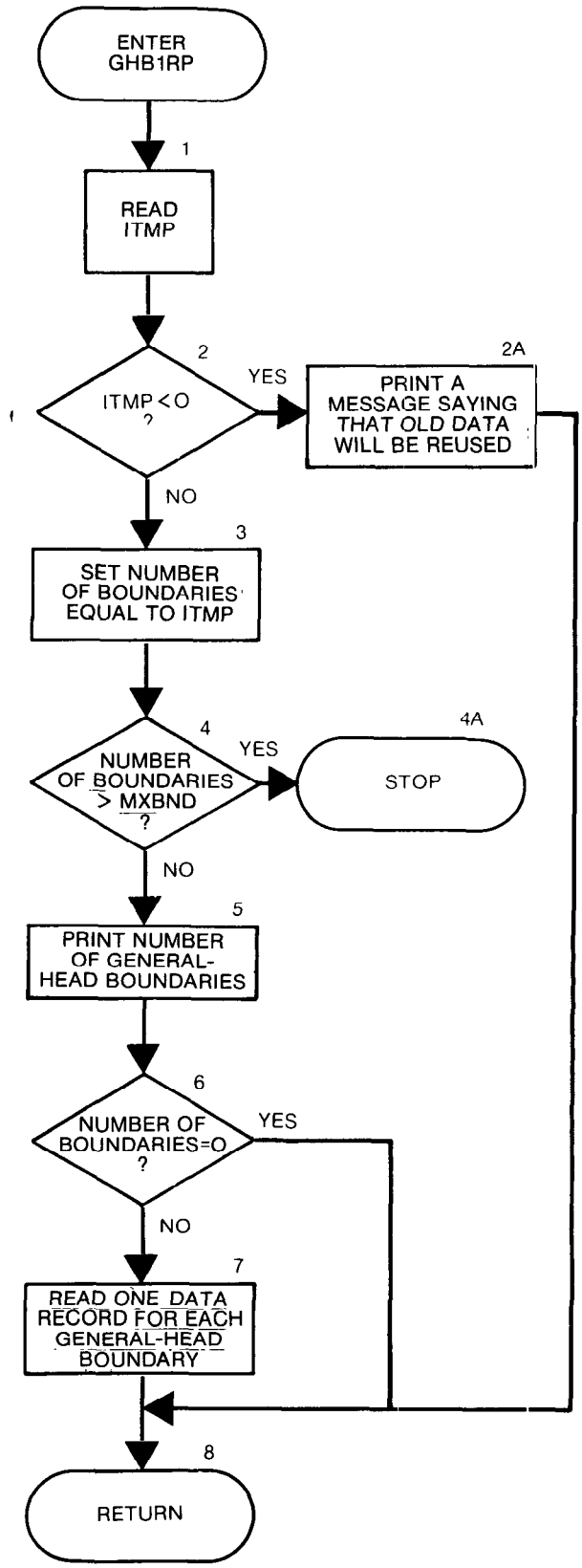
7. Read and print the layer, row, column, head, and conductance for each general-head boundary.

8. RETURN.

Flow Chart for Module GHB1RP

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

MXBND is the maximum number of general-head boundaries to be simulated.



```
      SUBROUTINE GHB1RP(BNDS,NBOUND,MXBND,IN,IOUT)
C
C
C-----VERSION 1651 02FEB1983 GHB1RP
C     ******************************************************************
C     READ DATA FOR GHB
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BNDS(5,MXBND)
C     ------------------------------------------------------------------
C
C1------READ ITMP(# OF GENERAL HEAD BOUNDS OR FLAG TO REUSE DATA.)
      READ(IN,8) ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD
      WRITE(IOUT,7)
    7 FORMAT(1HO,'REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS',
     1      ' PERIOD')
      GO TO 260
C
C3------IF ITMP=>0 THEN IT IS THE # OF GENERAL HEAD BOUNDS.
   50 NBOUND=ITMP
C
C4------IF MAX NUMBER OF BOUNDS IS EXCEEDED THEN STOP
      IF(NBOUND.LE.MXBND) GO TO 100
      WRITE(IOUT,99) NBOUND,MXBND
   99 FORMAT(1HO,'NBOUND(',I4,') IS GREATER THAN MXBND(',I4,')')
C
C4A-----ABNORMAL STOP
      STOP
C
C5------PRINT # OF GENERAL HEAD BOUNDS THIS STRESS PERIOD
  100 WRITE(IOUT,1) NBOUND
    1 FORMAT(1HO,//1X,I5,' HEAD-DEPENDENT BOUNDARY NODES')
C
C6------IF THERE ARE NO GENERAL HEAD BOUNDS THEN RETURN.
      IF(NBOUND.EQ.0) GO TO 260
C
C7------READ & PRINT DATA FOR EACH GENERAL HEAD BOUNDARY.
      WRITE(IOUT,3)
    3 FORMAT(1HO,15X,'LAYER',5X,'ROW',5X
     1,'COL   ELEVATION   CONDUCTANCE   BOUND NO.'/1X,15X,60('-'))
      DO 250 II=1,NBOUND
      READ (IN,4) K,I,J,BNDS(4,II),BNDS(5,II)
    4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,BNDS(4,II),BNDS(5,II),II
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4,I8)
      BNDS(1,II)=K
      BNDS(2,II)=I
      BNDS(3,II)=J
  250 CONTINUE
C
C8------RETURN
  260 RETURN
      END
```

List of Variables for Module GHB1RP

Variable	Range	Definition
BNDS	Package	DIMENSION (5,MXBND), Layer, row, column, head and conductance from boundary for each general-head boundary.
I	Module	Row number.
II	Module	Index for general-head boundaries.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ITMP	Module	Flag or number of boundaries. $\geq$ 0, number of bounds active during the current stress period. < 0, same bounds active during the last stress period will be active during the current stress period.
J	Module	Column number.
K	Module	Layer number.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current stress period.

Narrative for Module GHB1FM

This module adds terms representing riverhead boundaries to the accumulators HCOF and RHS.

1. If NBOUND is less than or equal to zero in the current stress period, there are no general-head boundaries. RETURN.

2. For each boundary in the BNDS list, DO STEPS 3-6.

3. Determine the column (IC), row (IR), and layer (IL).

4. If the cell is external (IBOUND(IC, IR, IL) $\leq$ 0), bypass processing on this boundary and go on to the next one.

5. If the cell is internal, get the boundary data (head and conductance).

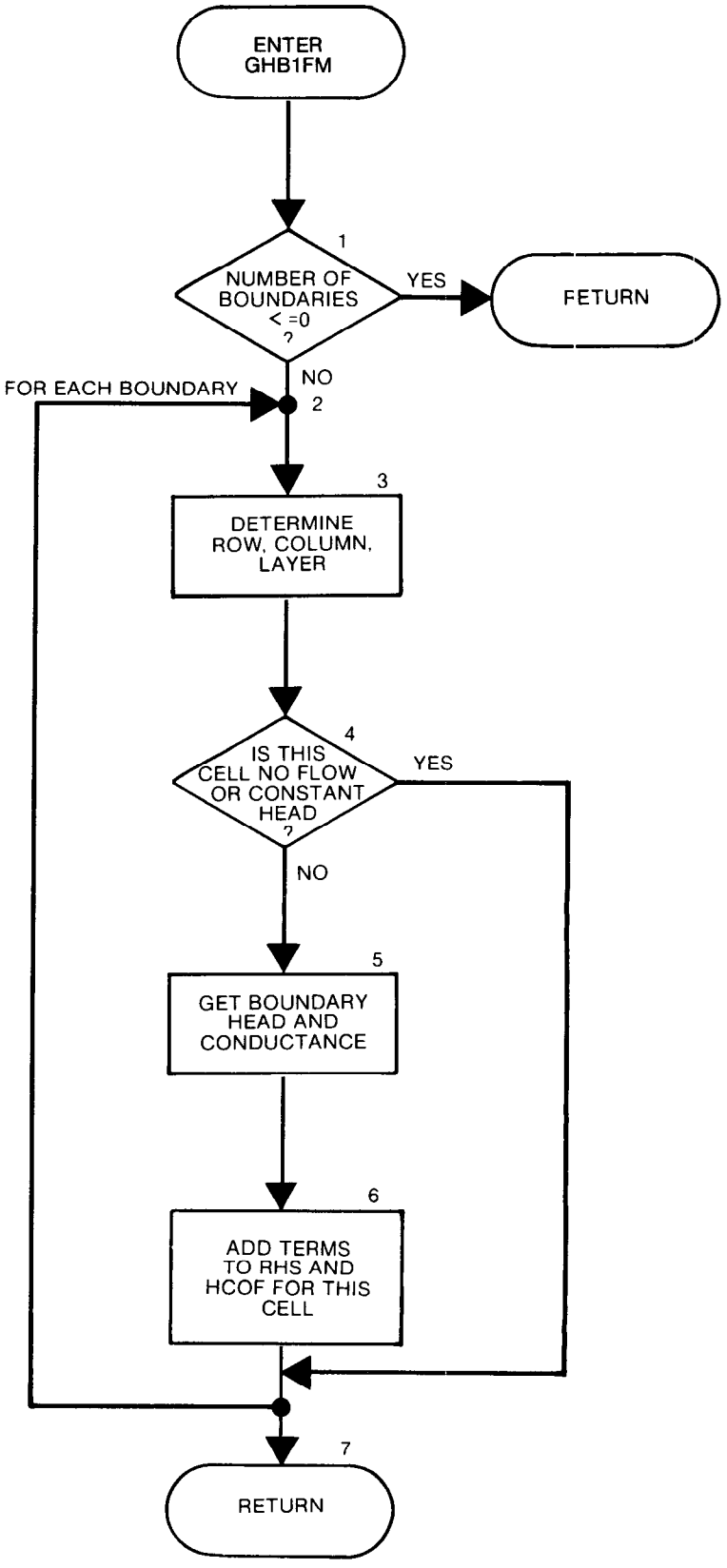
6. Add the -C*HB term (C is the conductance and HB is the boundary head) to the accumulator RHS and the term -C to the accumulator HCOF.

7. RETURN.

Flow Chart for Module GHB1FM

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

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



```
      SUBROUTINE GHB1FM(NBOUND,MXBND,BNDS,HCOF,RHS,IBOUND,
     1                  NCOL,NROW,NLAY)
C
C-----VERSION 1037 10APR1985 GHB1FM
C     ******************************************************************
C     ADD GHB TERMS TO RHS AND HCOF
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BNDS(5,MXBND),HCOF(NCOL,NROW,NLAY),
     1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NBOUND<=0 THEN THERE ARE NO GENERAL HEAD BOUNDS. RETURN.
      IF(NBOUND.LE.0) RETURN
C
C2------PROCESS EACH ENTRY IN THE GENERAL HEAD BOUND LIST (BNDS)
      DO 100 L=1,NBOUND
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING BOUNDARY
      IL=BNDS(1,L)
      IR=BNDS(2,L)
      IC=BNDS(3,L)
C
C4------IF THE CELL IS EXTERNAL THEN SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C5------SINCE THE CELL IS INTERNAL GET THE BOUNDARY DATA.
      HB=BNDS(4,L)
      C=BNDS(5,L)
C
C6------ADD TERMS TO RHS AND HCOF
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C
      RHS(IC,IR,IL)=RHS(IC,IR,IL)-C*HB
  100 CONTINUE
C
C7------RETURN
      RETURN
      END
```

List of Variables for Module GHB1FM

Variable	Range	Definition
BNDS	Package	DIMENSION (5,MXBND), Layer, row, column, head and conductance from boundary for each general-head boundary.
C	Module	Conductance from the external boundary.
HB	Module	Head on boundary.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
L	Module	Index for boundaries.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current stress period.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

Narrative for Module GHB1BD

This module calculates rates and volumes transferred between the aquifer and general-head boundaries.

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

2. If there are no general-head boundaries (NBOUND = 0), skip down to step 13 and put zeros into the budget terms for general-head boundaries.

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

4. For each general-head boundary, DO STEPS 5-13 accumulating flows from or into the general-head boundary.

5. Determine the row, column, and layer of the cell containing the general-head boundary.

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

7. Get the boundary parameters from the boundary list (BNDS).

8. Set RATE equal to the boundary conductance times the boundary head minus the head in the cell (RATE = C*(HB - HHNEW)).

9. If cell-by-cell flow terms are to be printed (IGHBCB < 0 and ICBCFL ≠ 0), print RATE.

10. If budget terms for individual cells are to be saved, add the RATE to the buffer (BUFF).

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

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

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

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

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

16. Increment the budget-term counter (MSUM). See the section in the Basic Package for a detailed explanation of VBVL, VBNM, and MSUM.

17. RETURN.

Flow Chart for Module GHB1BD

IBD is a flag which, if set, causes cell-by-cell flow terms for general-head boundary to be recorded.

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

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

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

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

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

C is the conductance between the boundary and the cell.

HB is the boundary head.

HHNEW is the head in the cell.

IGHBCB is a flag and a unit number.

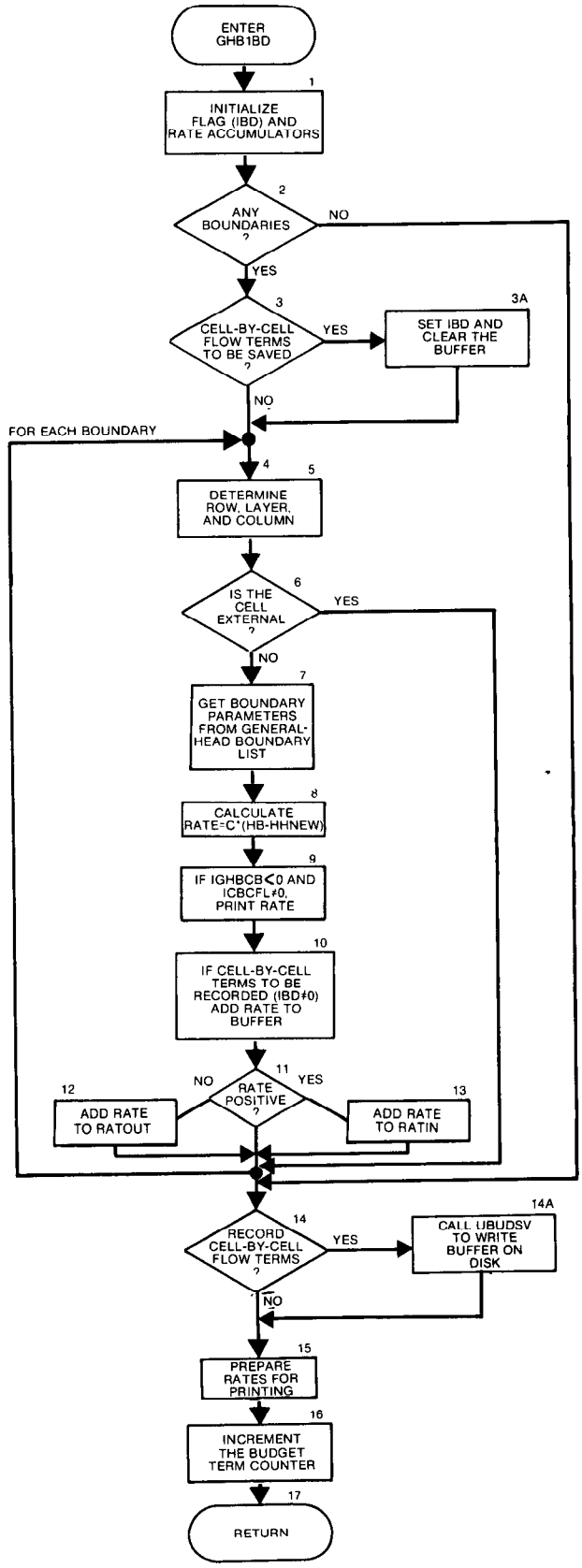
If IGHBCB > 0, it is the unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set.

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

If IGHBCB < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.

ICBCFL is a flag.

If ICBCFL ≠ 0, cell-by-cell flow terms will be either recorded or printed depending on IGHBCB for the current time step.



```
      SUBROUTINE GHB1BD(NBOUND,MXBND,VBNM,VBVL,MSUM,BNDS,DELT,HNEW,
     1    NCOL,NROW,NLAY,IBOUND,KSTP,KPER,IGHBCB,ICBCFL,BUFF,IOUT)
C
C-----VERSION 1612 12MAY1987 GHB1BD
C     ******************************************************************
C     CALCULATE VOLUMETRIC BUDGET FOR GHB
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM,TEXT
      DOUBLE PRECISION HNEW
      DIMENSION VBNM(4,MSUM),VBVL(4,MSUM),BNDS(5,MXBND),
     1          HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     2          BUFF(NCOL,NROW,NLAY)
      DIMENSION TEXT(4)
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /' HEA','D DE','P BO','UNDS'/
C     ------------------------------------------------------------------
C
C1------INITIALIZE CELL-BY-CELL FLOW TERM FLAG (IBD) AND
C1------ACCUMULATORS (RATIN AND RATOUT)
      IBD=0
      RATOUT=0.
      RATIN=0.
C
C2------IF NO BOUNDARIES THEN KEEP ZEROES IN ACCUMULATORS.
      IF(NBOUND.EQ.0) GO TO 200
C
C3------TEST TO SEE IF CELL-BY-CELL FLOW TERMS ARE NEEDED.
      IF(ICBCFL.EQ.0 .OR. IGHBCB.LE.0) GO TO 10
C
C3A-----SINCE CELL-BY-CELL FLOW TERMS ARE NEEDED CLEAR BUFFER & SET
C3A-----THE FLAG IBD.
      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 GENERAL HEAD BOUND ACCUMULATE FLOW INTO AQUIFER
   10 DO 100 L=1,NBOUND
C
C5------GET LAYER, ROW AND COLUMN OF EACH GENERAL HEAD BOUNDARY.
      IL=BNDS(1,L)
      IR=BNDS(2,L)
      IC=BNDS(3,L)
C
C6------IF CELL IS EXTERNAL THEN IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
```

```
C
C7------GET PARAMETERS FROM BOUNDARY LIST.
      HHNEW=HNEW(IC,IR,IL)
      HB=BNDS(4,L)
      C=BNDS(5,L)
C
C8------CALCULATE THE FOW RATE INTO THE CELL
      RATE=C*(HB-HHNEW)
C
C9------PRINT THE INDIVIDUAL RATES IF REQUESTED(IGHBCB<0).
      IF(IGHBCB.LT.0.AND.ICBCFL.NE.0) WRITE(IOUT,900) (TEXT(N),N=1,4),
     1    KPER,KSTP,L,IL,IR,IC,RATE
  900 FORMAT(1H0,4A4,'   PERIOD',I3,'   STEP',I3,'   BOUNDARY',I4,
     1    '   LAYER',I3,'   ROW',I4,'   COL',I4,'   RATE',G15.7)
C
C10-----IF CELL-BY-CELL TERMS ARE TO BE SAVED THEN PUT RATE IN BUFFER
      IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+RATE
C
C11-----SEE IF FLOW IS INTO AQUIFER OR OUT OF AQUIFER.
      IF(RATE)94,100,96
C
C12------FLOW IS OUT OF AQUIFER SUBTRACT RATE FROM RATOUT
   94 RATOUT=RATOUT-RATE
      GO TO 100
C
C13-----FLOW IS INTO AQIFER ADD RATE TO RATIN
   96 RATIN=RATIN+RATE
  100 CONTINUE
C
C14-----IF CELL-BY-CELL TERMS ARE TO BE SAVED THEN CALL
C14-----UTILITY MODULE UBUDSV
      IF(IBD.EQ.1) CALL UBUDSV(KSTP,KPER,TEXT,IGHBCB,BUFF,NCOL,NROW,
     1                         NLAY,IOUT)
C
C15-----MOVE RATES, VOLUMES AND LABELS INTO ARRAYS FOR PRINTING
  200 VBVL(3,MSUM)=RATIN
      VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
      VBVL(4,MSUM)=RATOUT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBNM(1,MSUM)=TEXT(1)
      VBNM(2,MSUM)=TEXT(2)
      VBNM(3,MSUM)=TEXT(3)
      VBNM(4,MSUM)=TEXT(4)
C
C16-----INCREMENT THE BUDGET TERM COUNTER
      MSUM=MSUM+1
C
C17-----RETURN
      RETURN
      END
```

List of Variables for Module GHB1BD

Variable	Range	Definition
BNDS	Package	DIMENSION (5,MXBND), Layer, row, column, head and conductance from the boundary for each general-head boundary.
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
C	Module	Conductance from the external boundary.
DELT	Global	Length of the current time step.
HB	Module	Head on boundary.
HHNEW	Module	HNEW (J,I,K), Single precision.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
IBD	Package	Flag. = 0, cell-by-cell flow terms for this package will not be recorded. ≠ 0, cell-by-cell flow terms for this package will be recorded.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
IC	Module	Index for columns.
IGHBCB	Package	Flag and a unit number. > 0, unit number on which cell-by-cell flow terms will be recorded whenever ICBCFL is set. = 0, cell-by-cell flow terms will not be printed or recorded. < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.
ICBCFL	Global	Flag. = 0, cell-by-cell flow terms will not be recorded or printed for the current time step. ≠ 0, cell-by-cell flow terms will be either printed or recorded (depending on IGHBCB) for the current time step.
IL	Module	Index for layers.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
L	Module	Index for general-head boundaries.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
MXBND	Package	Maximum number of head boundaries active at any one time.
NBOUND	Package	Number of head boundaries active during the current stress period.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.

List of Variables for Module GHB1BD (Continued)

Variable	Range	Definition
RATE	Module	Flow from a bound into a cell. (Reverse the sign to get flow into the bound.)
RATIN	Module	Accumulator for the total flow into the flow field out of the bounds.
RATOUT	Module	Accumulator for the total flow out of the flow field into the bounds.
TEXT	Module	Label to be printed or recorded with the array data.
VBNM	Global	DIMENSION (4,20), Labels for entries in the volumetric budget.
VBVL	Global	DIMENSION (4,20), Entries for the volumetric budget. For flow component N, the values in VBVL are: (1,N), Rate for the current time step into the flow field. (2,N), Rate for the current time step out of the flow field. (3,N), Volume into the flow field during simulation. (4,N), Volume out of the flow field during simulation.

CHAPTER 12

STRONGLY IMPLICIT PROCEDURE PACKAGE

Conceptualization and Implementation

General Theory

The discussion of the Strongly Implicit Procedure (SIP) presented here utilizes certain general concepts of matrix algebra and numerical analysis which may be reviewed in any standard reference, including those noted earlier by Peaceman (1977), Crichlow (1977) or Remson, Hornberger and Molz (1971). In addition to general background material, these three references provide descriptions of the Strongly Implicit Procedure itself which may be consulted to supplement the discussion presented here.

SIP is a method for solving a large system of simultaneous linear equations by iteration. The finite difference equation for a single cell, i,j,k, was shown in Chapter 2 to be of the form

$$\begin{aligned} &CV_{i,j,k-1/2}h_{i,j,k-1} + CC_{i-1/2,j,k}h_{i-1,j,k} + CR_{i,j-1/2,k}h_{i,j-1,} \\ &+ (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} \\ &- CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h_{i,j,k} \\ &+ CR_{i,j+1/2,k}h_{i,j+1,k} + CC_{i+1/2,j,k}h_{i+1,j,k} \\ &+ CV_{i,j,k+1/2}h_{i,j,k+1} = RHS_{i,j,k}. \end{aligned} \tag{79}$$

One equation of this form is written for each cell in the finite-difference grid, expressing the relationship among the heads at node i,j,k, and at each of the six adjacent nodes at the end of a time step. Because each equation may involve up to seven unknown values of head, and because the set of unknown head values changes from one equation to the next through the grid, the equations for the entire grid must be solved simultaneously at each time step. The solution consists of one value of head for each node, for the end of the step.

The discussion of the SIP procedure presented here is based on the notation of Weinstein, Stone and Kwan (1969), the developers of SIP. Using their notation, equation (79) may be written.

$$Z_{i,j,k}h_{i,j,k-1} + B_{i,j,k}h_{i-1,j,k} + D_{i,j,k}h_{i,j-1,k} + E_{i,j,k}h_{i,j,k} + F_{i,j,k}h_{i,j+1,k} + H_{i,j,k}h_{i+1,j,k} + S_{i,j,k}h_{i,j,k+1} = Q_{i,j,k}. \quad (80)$$

The coefficients in equation (80) all are labelled with the index i,j,k to show that they are associated with the equation for node i,j,k. Thus $Z_{i,j,k}$ of equation (80) is equivalent to $CL_{i,j,k-1/2}$ of equation (79); $E_{i,j,k}$ of equation (80) is equivalent to the expression $(-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2k} - CR_{i,j+1/2k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})$ of equation (79); and so on.

As pointed out in Chapter 2, the entire set of equations of the form of (80) can be summarized in matrix form as

$$[A]\{h\} = \{q\} \quad (81)$$

where [A] is the matrix of coefficients of head, {h} is a vector of head values, and {q} is a vector of the right-hand terms of equation (80). Figure 46 shows the elements of the coefficient matrix and of the two vectors for a mesh of three rows, four columns and two layers. Notice that the matrix [A] is sparse--i.e., that there are very few nonzero elements--and that these are all located on just seven diagonals, as indicated in figure 47.

Examination of equations (79) and (80) will show that the term $CL_{i,j,k-1/2}$ of equation (79) appears both as the coefficient Z in equation (80) for node i,j,k, and as the coefficient S in the corresponding equation for node i,j,k-1, that is

$$Z_{i,j,k} = S_{i,j,k-1} \quad (82)$$

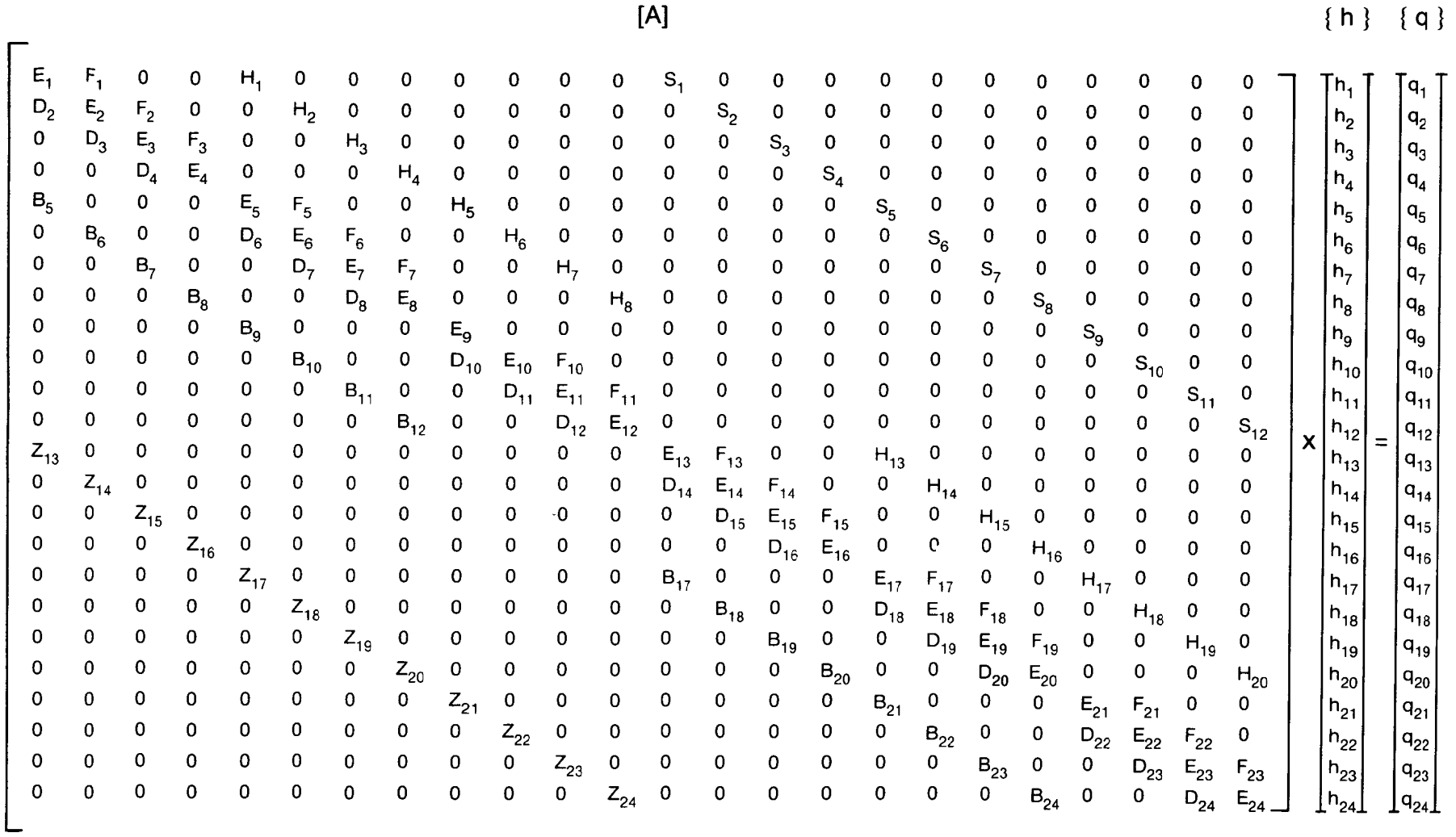
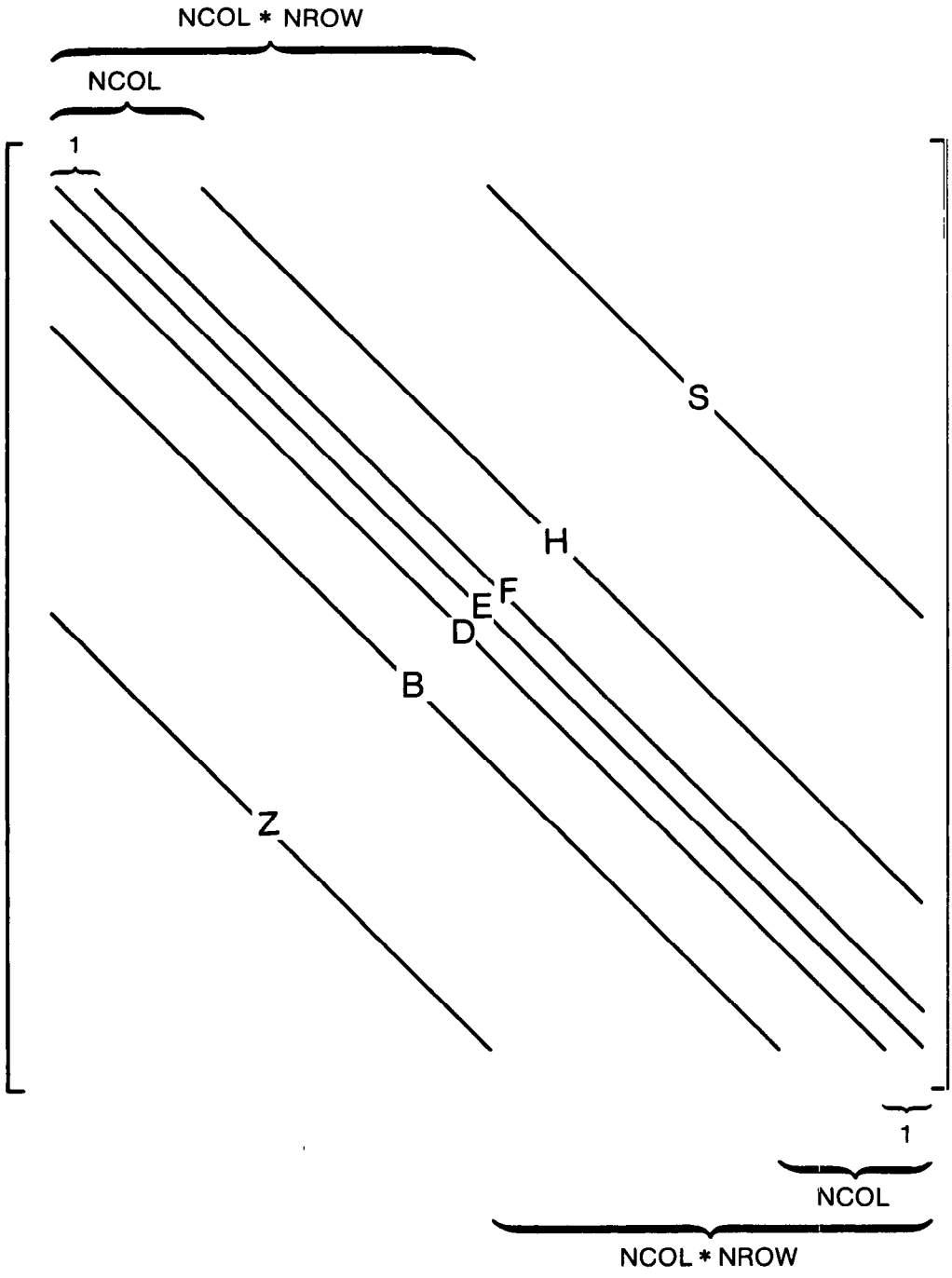


Figure 46.—Correspondence between the finite-difference equations and the matrix equation for a grid of three rows, four columns, and two layers.



Brackets indicate horizontal spacing, in matrix columns, between nonzero diagonals (e.g., diagonals E and F are adjacent).

Figure 47.—Structure of coefficient matrix showing nonzero diagonals.

Similarly,

$$B_{i,j,k} = H_{i-1,j,k} \tag{83}$$

and

$$D_{i,j,k} = F_{i,j-1,k}. \tag{84}$$

Replacing each Z, B, and D coefficient in the matrix of Figure 46 with the equivalent S, H, or F element, as defined by equations (82) - (84), yields the matrix of Figure 48, which is readily seen to be symmetric. Thus the coefficient matrix [A] of equation (81) is symmetric as well as sparse.

A system of equations of the form of (81) can be solved by direct methods if [A] can be factored into two matrices [L*] and [U*], such that [L*] is in lower triangular form (all nonzero elements are on or below the main diagonal), while [U*] is in upper triagular form (all nonzero elements are on or above the main diagonal), and all elements on the main diagonal of [U*] are equal to one. Figure (49) illustrates the characteristics of [L*] and [U*] relative to [A] for a 3 x 3 matrix [A]. Once this factoring has been accomplished, a technique known as "backward and forward substitution" can be used to complete the solution. However, a difficulty arises in that, even though [A] is a sparse matrix, [L*] and [U*] are generally not sparse, and a great deal of computer memory and time may be needed to calculate all of their nonzero elements. In addition, round-off errors may become unacceptably large.

The Strongly Implicit Procedure seeks to find a matrix [B] such that the sum matrix [A + B] can be factored easily into two matrices [L] and [U], where [A + B], [L], and [U] meet the following conditions:

(1) [A + B] is "close" to [A];

(2) [L] is in lower triangular form while [U] is in upper triangular form, and all entries along the main diagonal of [U] are equal to unity;

$E_{1,1,1}$	$F_{1,1,1}$	0	$H_{1,1,1}$	0	0	$S_{1,1,1}$	0	0	0	0	0
$F_{1,1,1}$	$E_{1,2,1}$	$F_{1,2,1}$	0	$H_{1,2,1}$	0	0	$S_{1,2,1}$	0	0	0	0
0	$F_{1,2,1}$	$E_{1,3,1}$	0	0	$H_{1,3,1}$	0	0	$S_{1,3,1}$	0	0	0
$H_{1,1,1}$	0	0	$E_{2,1,1}$	$F_{2,1,1}$	0	0	0	0	$S_{2,1,1}$	0	0
0	$H_{1,2,1}$	0	$F_{2,1,1}$	$E_{2,2,1}$	$F_{2,2,1}$	0	0	0	0	$S_{2,2,1}$	0
0	0	$H_{1,3,1}$	0	$F_{2,2,1}$	$E_{2,3,1}$	0	0	0	0	0	$S_{2,3,1}$
$S_{1,1,1}$	0	0	0	0	0	$E_{1,1,2}$	$F_{1,1,2}$	0	$H_{1,1,2}$	0	0
0	$S_{1,2,1}$	0	0	0	0	$F_{1,1,2}$	$E_{1,2,2}$	$F_{1,2,2}$	0	$H_{1,2,2}$	0
0	0	$S_{1,3,1}$	0	0	0	0	$F_{1,2,2}$	$E_{1,3,2}$	0	0	$H_{1,3,2}$
0	0	0	$S_{2,1,1}$	0	0	$H_{1,1,2}$	0	0	$E_{2,1,2}$	$F_{2,1,2}$	0
0	0	0	0	$S_{2,2,1}$	0	0	$H_{1,2,2}$	0	$F_{2,1,2}$	$E_{2,2,2}$	$F_{2,2,2}$
0	0	0	0	0	$S_{2,3,1}$	0	0	$H_{1,3,2}$	0	$F_{2,2,2}$	$E_{2,3,2}$

Figure 48.—Symmetric coefficient matrix for a grid containing two rows, three columns, and two layers.

$$
\underset{[A]}{\begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 2 \\ 3 & 2 & -2 \end{bmatrix}} \underset{\{h\}}{\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}} = \underset{\{q\}}{\begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}}
$$

$$
\underset{[L^*]}{\begin{bmatrix} 1 & 0 & 0 \\ -1 & 3 & 0 \\ 3 & -4 & -1 \end{bmatrix}} \underset{[U^*]}{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}} \underset{\{h\}}{\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}} = \underset{\{q\}}{\begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}}
$$

Figure 49.—Decomposition of a coefficient matrix into lower and upper triangular matrices.

(3) [L] and [U] are both sparse; and

(4) both [L] and [U] have just four nonzero diagonals.

Suppose a matrix [B] is constructed in an attempt to satisfy these conditions; the term [B] {h} can be added to each side of equation (81) to give

$$[A + B]\{h\} = \{q\} + [B]\{h\} \tag{85}$$

A solution vector {h} for equation (85) must also be a solution vector for equation (81). The presence of the vector {h} on both sides of equation (85) presents an immediate difficulty; however, if an iterative approach to the solution is utilized (chapter 2), values of h from the preceding iteration may be used in the head vector on the right. That is, equation (85) may be expressed in the form

$$[A + B]\{h^{\ell}\} = \{q\} + [B]\{h^{\ell-1}\} \tag{86}$$

where $\{h^{\ell}\}$ is a vector of head values from iteration ℓ, and $\{h^{\ell-1}\}$ a vector of head values from iteration ℓ-1. In equation (86), $\{h^{\ell-1}\}$ is actually used as an approximation to $\{h^{\ell}\}$. If the matrix [B] were known, solution of (86) would be straightforward; for according to the properties postulated above, [A + B] could be factored easily into the sparse matrices [L] and [U], allowing the use of forward and backward substitution. Thus the problem of solving equation (86) is equivalent to that of finding an appropriate matrix [B]. In practice, however, the solution is pursued in terms of the matrices [A], [A + B], [L] and [U]. The term $[A + B]\{h^{\ell-1}\}$ is subtracted from each side of (86) to yield

$$[A + B]\{h^{\ell}\} - [A + B]\{h^{\ell-1}\} = \{q\} - [A]\{h^{\ell-1}\} \tag{87}$$

or

$$[A + B]\{h^{\ell} - h^{\ell-1}\} = \{q\} - [A]\{h^{\ell-1}\} \tag{88}$$

In order that the conditions specified above for [L], [U], and [A + B] may be satisified, [A + B] must contain six nonzero diagonals which were not present in [A], as shown in figure 50; the effect of these additional nonzero diagonals is to introduce new terms into the equation for node i,j,k, involving heads at nodes not adjacent to i,j,k. The relationship between the elements of [A + B] and the elements of [L] and [U] is as given in the following equations, where as indicated in figures 50 and 51, a, b, c, and d, refer to elements of [L], e, f, and g, refer to elements of [U] above the main diagonal, and capital letters refer to elements of [A + B].

$$Z'_{i,j,k} = a_{i,j,k} \tag{89-a}$$

$$A'_{i,j,k} = a_{i,j,k}e_{i,j,k-1} \tag{89-b}$$

$$T'_{i,j,k} = a_{i,j,k}f_{i,j,k-1} \tag{89-c}$$

$$B'_{i,j,k} = b_{i,j,k} \tag{89-d}$$

$$C'_{i,j,k} = e_{i-1,j,k}b_{i,j,k} \tag{89-e}$$

$$D'_{i,j,k} = c_{i,j,k} \tag{89-f}$$

$$E'_{i,j,k} = a_{i,j,k}g_{i,j,k-1} + b_{i,j,k}f_{i-1,j,k} + e_{i,j-1,k}c_{i,j,k}+d_{i,j,k} \tag{89-g}$$

$$F'_{i,j,k} = d_{i,j,k}e_{i,j,k} \tag{89-h}$$

$$G'_{i,j,k} = f_{i,j-1,k}c_{i,j,k} \tag{89-i}$$

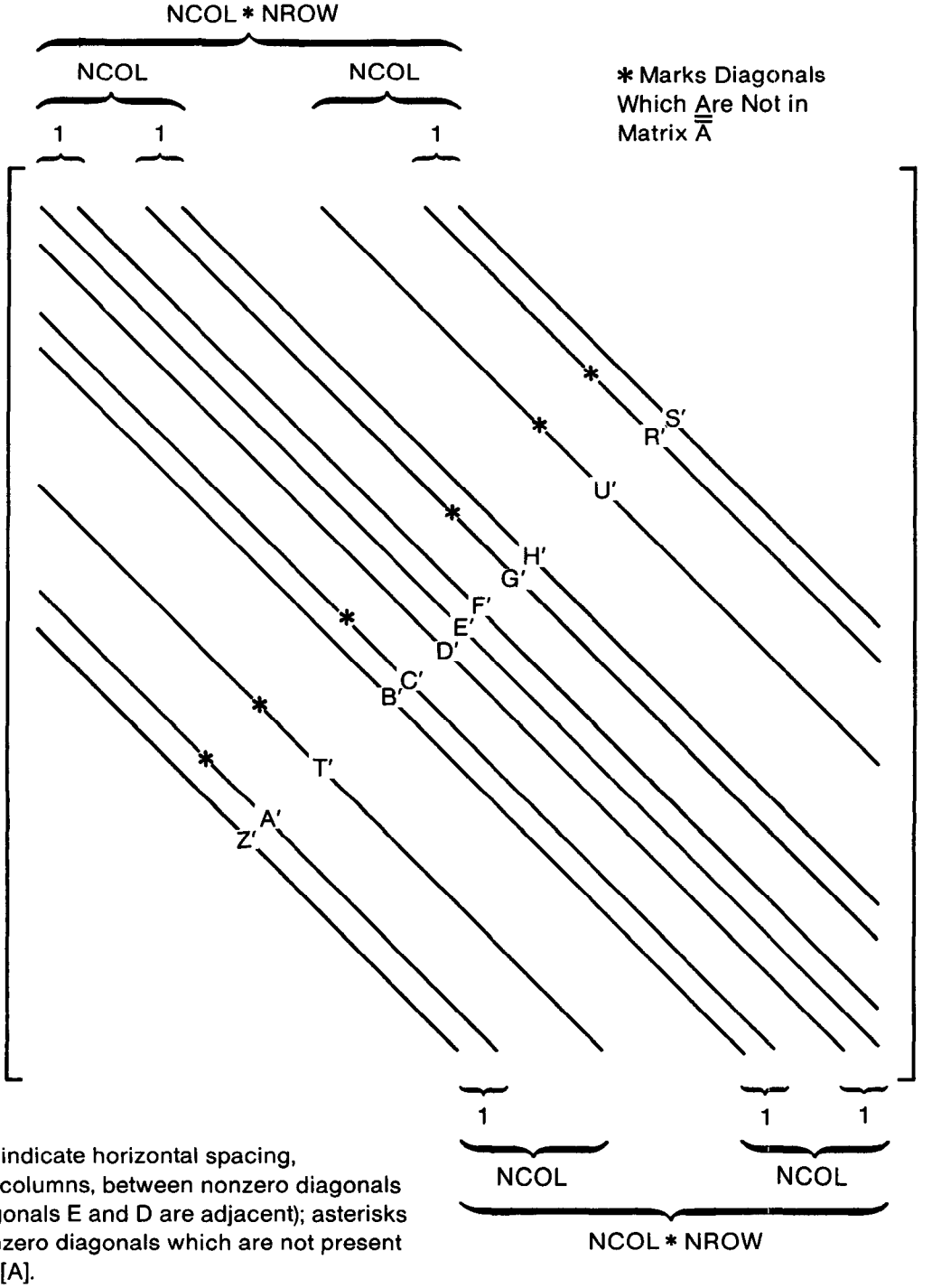
$$H'_{i,j,k} = f_{i,j,k}d_{i,j,k} \tag{89-j}$$

$$U'_{i,j,k} = b_{i,j,k}g_{i-1,j,k} \tag{89-k}$$

$$R'_{i,j,k} = g_{i,j-1,k}c_{i,j,k} \tag{89-l}$$

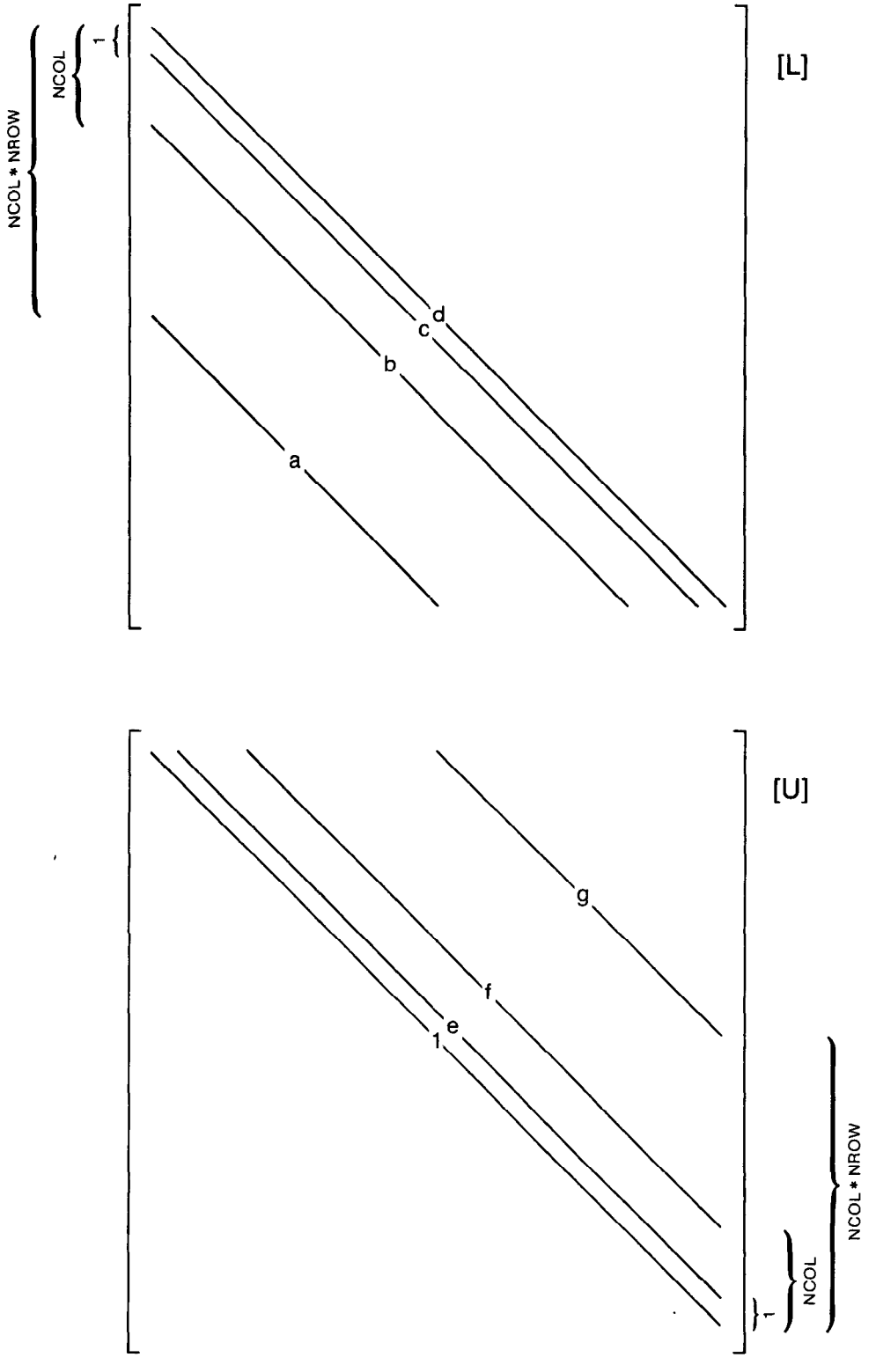
$$S'_{i,j,k} = g_{i,j,k}d_{i,j,k} \tag{89-m}$$

If the subscript of an element in equations (89-a...m) places the element outside of the grid boundary, the element is assumed to be equal to zero. The 13 equations contain 20 unknown values, the elements of [L], [U], and



Brackets indicate horizontal spacing, in matrix columns, between nonzero diagonals (e.g., diagonals E and D are adjacent); asterisks mark nonzero diagonals which are not present in matrix [A].

Figure 50.—Structure of matrix [A+B] showing nonzero diagonals.



Brackets indicate vertical spacing, in matrix rows, between nonzero diagonals (e.g., diagonals d and e are adjacent).

Figure 51.—Structure, showing nonzero diagonals, of the lower triangular factor [L] and the upper triangular factor [U] of matrix [A+B].

[A + B]. This indicates that there are many matrices [B] which can be added to [A] so that the sum can be factored into upper and lower triangular matrices of the form of [L] and [U]. However, the requirement that [A + B] must be "close" to [A], or equivalently that

$$[A + B]\{h\} \approx [A]\{h\} \tag{90}$$

has not yet been used. In terms of the elements of [A + B] {h} and [A] {h} associated with an individual node, i,j,k, equation (90) implies that

$$\begin{aligned}
&Z'_{i,j,k}h_{i,j,k-1} + A'_{i,j,k}h_{i,j+1,k-1} + T'_{i,j,k}h_{i+1,j,k-1} \\
&+ B'_{i,j,k}h_{i-1,j,k} + C'_{i,j,k}h_{i-1,j+1,k} + D'_{i,j,k}h_{i,j-1,k} \\
&+ E'_{i,j,k}h_{i,j,k} + F'_{i,j,k}h_{i,j+1,k} + G'_{i,j,k}h_{i+1,j-1,k} \\
&+ H'_{i,j,k}h_{i+1,j,k} + U'_{i,j,k}h_{i-1,j,k+1} + R'_{i,j,k}h_{i,j-1,k+1} \\
&+ S'_{i,j,k}h_{i,j,k+1} \approx Z_{,j,k}h_{i,j,k-1} + B_{i,j,k}h_{i-1,j,k} \\
&+ D_{i,j,k}h_{i,j-1,k} + E_{i,j,k}h_{i,j,k} + F_{i,j,k}h_{i,j+1,k} + H_{i,j,k}h_{i+1,j,k} \\
&+ S_{i,j,k}h_{i,j,k+1}
\end{aligned} \tag{91}$$

Equation (91) can be rearranged so that the terms from the six nonzero diagonals not present in [A] are all on the right side, while the left side is made up of differences between elements of matrix [A] and corresponding elements of matrix [A + B], i.e.

$$\begin{aligned}
&(Z_{i,j,k}-Z'_{i,j,k})h_{i,j,k-1} + (B_{i,j,k}-B'_{i,j,k})h_{i-1,j,k} \\
&+ (D_{i,j,k}-D'_{i,j,k})h_{i,j-1,k} + (E_{i,j,k}-E'_{i,j,k})h_{i,j,k} \\
&+ (F_{i,j,k}-F'_{i,j,k})h_{i,j+1,k} + (H_{i,j,k}-H'_{i,j,k})h_{i+1,j,k} \\
&+ (S_{i,j,k}-S'_{i,j,k})h_{i,j,k+1} \approx A'_{i,j,k}h_{i,j+1,k-1} \\
&+ T'_{i,j,k}h_{i+1,j,k-1} + C'_{i,j,k}h_{i-1,j+1,k} + G'_{i,j,k}h_{i+1,j-1,k} \\
&+ U'_{i,j,k}h_{i-1,jk+1} + R'_{i,j,k}h_{i,j-1,k+1}
\end{aligned} \tag{92}$$

The terms on the right side of (92), corresponding to the six nonzero diagonals of [A + B] not appearing in [A], are all derived from the matrix B, and all involve the heads at nodes not adjacent to node, i,j,k; by contrast,

the terms on the left side of (92) are derived from both [A] and [B], and involve the heads at i,j,k and the six adjacent nodes.

To reduce the effect of the terms corresponding to nodes not adjacent to i,j,k, three parameters, here termed α, β and γ, and all chosen between zero and one, are introduced as multipliers of the terms on the right side of equation (92). Ultimately, as the solution of the matrix equations ((85) or (86)) is implemented, these multipliers take on the role of iteration parameters. They are brought into equation (92) as follows:

$$(Z_{i,j,k}-Z'_{i,j,k})h_{i,j,k-1} + (B_{i,j,k}-B'_{i,j,k})h_{i-1,j,k}$$
$$+ (D_{i,j,k}-D'_{i,j,k})h_{i,j-1,k} + (E_{i,j,k}-E'_{i,j,k})h_{i,j,k}$$
$$+ (F_{i,j,k}-F'_{i,j,k})h_{i,j+1,k} + (H_{i,j,k}-H'_{i,j,k})h_{i+1,j,k}$$
$$+ (S_{i,j,k}-S'_{i,j,k})h_{i,j,k+1} \approx \alpha A'_{i,j,k}h_{i,j+1,k-1}$$
$$+ \beta T'_{i,j,k}h_{i+1,j,k-1} + \gamma C'_{i,j,k}h_{i-1,j+1,k}$$
$$+ \gamma G'_{i,j,k}h_{i+1,j-1,k} + \beta U'_{i,j,k}h_{i-1,jk+1}$$
$$+ \alpha R'_{i,j,k}h_{i-1,k+1} \qquad (93)$$

Next the heads on right side of (93), corresponding to nodes not adjacent to i,j,k, are expressed in terms of heads at nodes which are adjacent to i,j,k. This is done by noting that, for example, node i, j+1, k-1 lies at the corner of a rectangle, the other three corners of which are: i,j,k-1; i,j+1,k; and i,j,k. Thus using the rules in interpolation illustrated in figure 52, $h_{i,j+1,k-1}$ is given approximately by

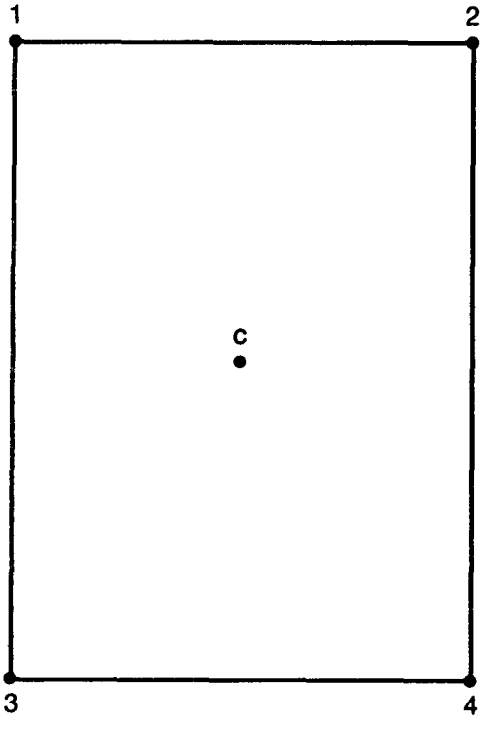
$$h_{i,j+1,k-1} = h_{i,j+1,k} + h_{i,j,k-1} - h_{i,j,k}. \qquad (94\text{-a})$$

Similarly,

$$h_{i+1,j,k-1} = h_{i,j,k-1} + h_{i+1,j,k} - h_{i,j,k} \qquad (94\text{-b})$$

$$h_{i-1,j+1,k} = h_{i-1,j,k} + h_{i,j+1,k} - h_{i,j,k} \qquad (94\text{-c})$$

Suppose the Function f Is Known at 2, 3 and 4.



By interpolation the Function
at the Center Can be Approximated by

$$f_1(c) \approx \frac{f(2) + f(3)}{2}$$

and

$$f_2(c) \approx \frac{f(1) + f(4)}{2}$$

Suppose

$$f_1(c) \approx f_2(c)$$

Then

$$\frac{f(2) + f(3)}{2} \approx \frac{f(1) + f(4)}{2}$$

Therefore

$$f(1) \approx f(2) + f(3) - f(4)$$

Figure 52.—Estimation of a function at one corner of a rectangle in terms of the values of the function at the other three corners.

$$h_{i+1,j-1,k} = h_{i+1,j,k} + h_{i,j-1,k} - h_{i,j,k} \qquad (94\text{-d})$$

$$h_{i-1,j,k+1} = h_{i,j,k+1} + h_{i-1,j,k} - h_{i,j,k} \qquad (94\text{-e})$$

$$h_{i,j-1,k+1} = h_{i,j,k+1} + h_{i,j-1,k} - h_{i,j,k}. \qquad (94\text{-f})$$

Substituting equations (94-a...f) into equation (93) and reorganizing gives

$$\begin{aligned}
&(Z'_{i,j,k} - Z_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k})\, h_{i,j,k-1} \\
&+ (B'_{i,j,k} - B_{i,j,k} + \gamma C'_{i,j,k} + \beta U'_{i,j,k})\, h_{i-1,j,k} \\
&+ (D'_{i,j,k} - D_{i,j,k} + \gamma G'_{i,j,k} + \alpha R'_{i,j,k})\, h_{i,j-1,k} \\
&+ (E'_{i,j,k} - E_{i,j,k} - \alpha A'_{i,j,k} - \beta T'_{i,j,k} - \gamma C'_{i,j,k} \\
&\quad - \gamma G'_{i,j,k} - \beta U'_{i,j,k} - \alpha R'_{i,j,k})\, h_{i,j,k} \\
&+ (F'_{i,j,k} - F_{i,j,k} + \alpha A'_{i,j,k} + \gamma C'_{i,j,k})\, h_{i,j+1,k} \\
&+ (H'_{i,j,k} - H_{i,j,k} + \beta T'_{i,j,k} + \gamma G'_{i,j,k}) h_{i+1,j,k} \\
&+ (S'_{i,j,k} - S_{i,j,k} + \beta U'_{i,j,k} + \alpha R'_{i,j,k}) h_{i,j,k+1} \approx 0 \qquad (95)
\end{aligned}$$

The relation expressed in equation (95) can be satisfied if each coefficient is approximately equal to zero. Setting these coefficients equal to zero yields the equations

$$Z'_{i,j,k} - Z_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k} = 0 \qquad (96\text{-a})$$

$$B'_{i,j,k} - B_{i,j,k} + \gamma C'_{i,j,k} + \beta U'_{i,j,k} = 0 \qquad (96\text{-b})$$

$$D'_{i,j,k} - D_{i,j,k} + \gamma G'_{i,j,k} + \alpha R'_{i,j,k} = 0 \qquad (96\text{-c})$$

$$\begin{aligned}
E'_{i,j,k} - E_{i,j,k} &- \alpha A'_{i,j,k} - \beta T'_{i,j,k} \\
&- \gamma C'_{i,j,k} - \gamma G'_{i,j,k} \\
&- \beta U'_{i,j,k} - \alpha R'_{i,j,k} = 0
\end{aligned} \qquad (96\text{-d})$$

$$F'_{i,j,k} - F_{i,j,k} + \alpha A'_{i,j,k} + \gamma C'_{i,j,k} = 0 \qquad (96\text{-e})$$

$$H'_{i,j,k} - H_{i,j,k} + \beta T'_{i,j,k} + \gamma G'_{i,j,k} = 0 \qquad (96\text{-f})$$

$$S'_{i,j,k} - S_{i,j,k} + \beta U'_{i,j,k} + \alpha R'_{i,j,k} = 0 \qquad (96\text{-g})$$

Equations (96-a...g) and (89-a...m) form a system of 20 equations in 20 unknowns which when solved, will yield the entries of [A + B], [L] and [U] such that [A + B] is "close" to [A], and can be readily factored into

[L] and [U], where [L] and [U] are both sparse and have the required lower triangular and upper triangular forms. For example, substituting equations (89-a, -b, and -c) into equation (96-a) and rearranging yields

$$a_{i,j,k} = Z_{i,j,k}/(1 + \alpha e_{i,j,k-1} + \beta f_{i,j,k-1}). \tag{97-a}$$

Similarly,

$$b_{i,j,k} = B_{i,j,k}/(1 + \gamma e_{i-1,j,k} + \beta g_{i-1,j,k}) \tag{97-b}$$

$$c_{i,j,k} = D_{i,j,k}/(1 + \gamma f_{i,j-1,k} + \alpha g_{i,j-1,k}) \tag{97-c}$$

$$A'_{i,j,k} = a_{i,j,k} e_{i,j,k-1} \tag{97-d}$$

$$C'_{i,j,k} = e_{i-1,j,k} b_{i,j,k} \tag{97-e}$$

$$G'_{i,j,k} = f_{i,j-1,k} c_{i,j,k} \tag{97-f}$$

$$R'_{i,j,k} = g_{i,j-1,k} c_{i,j,k} \tag{97-g}$$

$$T'_{i,j,k} = a_{i,j,k} f_{i,j,k-1} \tag{97-h}$$

$$U'_{i,j,k} = b_{i,j,k} g_{i-1,j,k} \tag{97-i}$$

$$\begin{aligned} d_{i,j,k} = {} & E_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k} \\ & + \gamma C'_{i,j,k} + \gamma G'_{i,j,k} + \beta U'_{i,j,k} \\ & + \alpha R'_{i,j,k} - a_{i,j,k} g_{i,j,k-1} - b_{i,j,k} f_{i-1,j,k} \\ & - e_{i,j-1,k} c_{i,j,k} \end{aligned} \tag{97-j}$$

$$e_{i,j,k} = (F_{i,j,k} - \alpha A'_{i,j,k} - \gamma C'_{i,j,k})/d_{i,j,k} \tag{97-k}$$

$$f_{i,j,k} = (H_{i,j,k} - \beta T'_{i,j,k} - \gamma G'_{i,j,k})/d_{i,j,k} \tag{97-l}$$

$$g_{i,j,k} = (S_{i,j,k} - \alpha R'_{i,j,k} - \beta U'_{i,j,k})/d_{i,j,k} \tag{97-m}$$

Using these relations to provide the elements of [L] and [U], [A + B] may be replaced with the product [L][U] in (88) to yield

$$[L][U] \{h^{\ell} - h^{\ell-1}\} = \{q\} - [A] \{h^{\ell-1}\} \tag{98}$$

where again the superscript ℓ refers to the current iteration level, and ℓ-1 to the preceding iteration level. We next define the vector $\{RES^{\ell}\}$ by

$$\{RES^{\ell}\} = \{q\} - [A]\{h^{\ell-1}\} \quad (99)$$

Using this notation equation (98) can be written

$$[L][U]\{h^{\ell}-h^{\ell-1}\} = \{RES^{\ell}\} \quad (100)$$

Equation (100) can now be solved by a process of forward and backward substitution. The first step involves forward substitution to solve for the vector $\{v\}$ in the equation

$$[L]\{v\} = \{RES^{\ell}\} \quad (101)$$

where $\{v\} = [U]\{h^{\ell}-h^{\ell-1}\}$. The vector $\{v\}$ determined in this way is then ultilized in a process of back substitution to solve for the vector $\{h^{\ell}-h^{\ell-1}\}$ in the equation

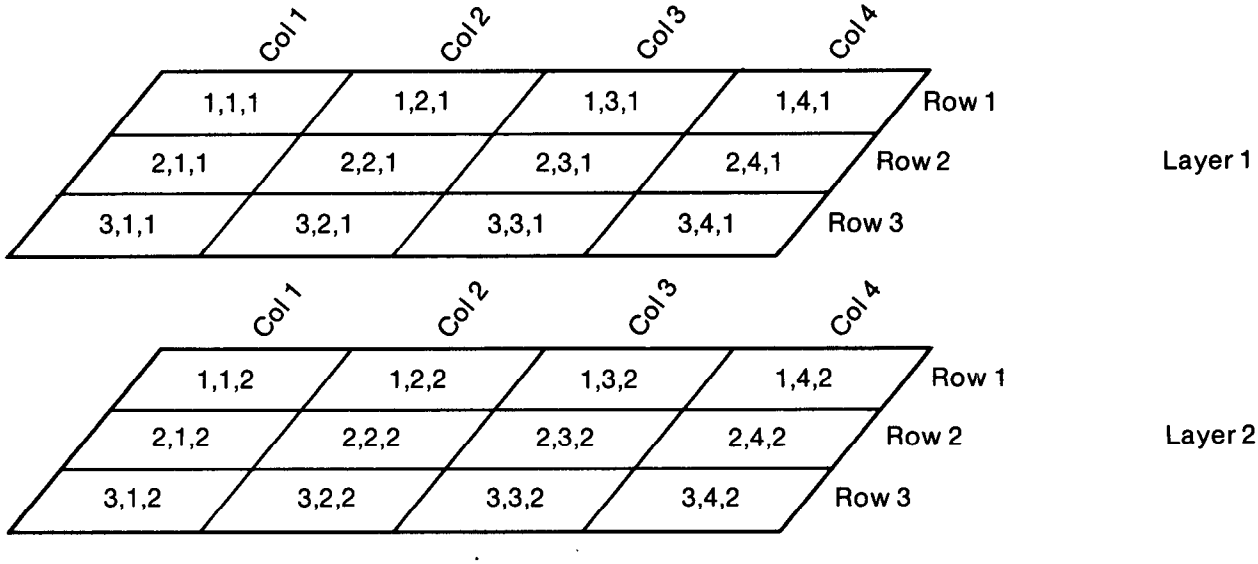
$$[U]\{h^{\ell}-h^{\ell-1}\} = \{v\} \quad (102)$$

In earlier discussions, the coefficients of the equations and hence the elements of the matrices were identified by the indices of the cells, as shown in figure 53-a. To illustrate the process of forward substitution, used to calculate the elements of the vector $\{v\}$, it is convenient to renumber the equations sequentially using a single index, as shown in figure 53-b. Because all elements in [L] above the main diagonal are zero, the first linear equation represented by matrix equation (101) is

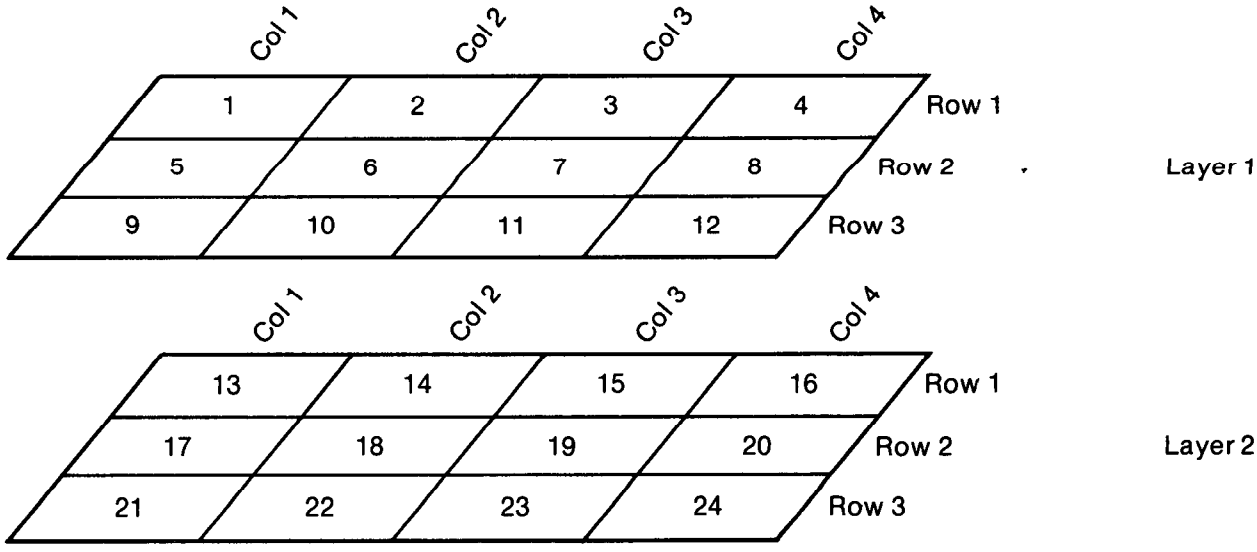
$$d_1 v_1 = RES_1^{\ell} \quad (103)$$

In equation (103), the term d_1 has been determined through equation (97-j), and RES_1^{ℓ} has been calculated through equation (99) as an element of the vector (RES^{ℓ}); thus (103) can be solved immediately for the value of v_1. The second equation represented by matrix equation (101) is

$$c_2 v_1 + d_2 v_2 = RES_2^{\ell} \quad (104)$$



(a) Cell Numbering With 3 Indices



(b) Cell Numbering With 1 Index

Figure 53.—Cell numbering schemes for a grid using three indices and using one index.

Again, c_2 and d_2 are known from equations (97), and RES_2^{ℓ} is known from equation (99); using the value of v_1 from the solution of equation (103), (104) can be solved for v_2.

The general equation for an element of $\{v\}$ has the form

$$v_n = (RES_n^{\ell} - a_n v_{n-NRC} - b_n v_{n-NCOL} - c_n v_{n-1})/d_n \qquad (105)$$

where NRC is the number of cells in the layer, NCOL is the number of columns in the model, the coefficeints a_n, b_n, c_n and d_n are all determined through equations (97) and RES_n^{ℓ} is determined through equation (99). The terms a_n and b_n are zero for the first and second eqations ((103) and (104)); and each equation involves elements of $\{v\}$ determined earlier in the sequence. This procedure of forward substitution, in which the elements of $\{v\}$ are determined in sequence, is possible because of the lower triangular form of the matrix [L]--i.e., because [L] has only zeros to the right of the main diagonal.

Back substitution is next used to calculate the elements of the vector $\{h^{\ell} - h^{\ell-1}\}$ from the elements of $\{v\}$, thus solving equation (102). The process of back substitution is similar to that of forward substitution except that, because the matrix [U] is upper triangular, the order of calculation is reversed. When the vector $\{h^{\ell} - h^{\ell-1}\}$ has been calculated, it is added to the vector $\{h^{\ell-1}\}$ to obtain $\{h^{\ell}\}$, the vector of head values corresponding to iteration ℓ.

In summary, the problem of solving the equation

$$[A]\{h\} = \{q\} \qquad (106)$$

has thus been converted into an iterative process in which: (1) the matrices [L] and [U] are determined using equations (97); (2) the vector $\{RES^{\ell}\}$ is calculated using the vector $\{q\}$, the matrix [A] and heads from the preceding iteration; (3) equation (100) is then solved using forward

and backward substitution to obtain the vector $\{h^{\ell}-h^{\ell-1}\}$; and (4) the vector $\{h^{\ell-1}\}$ is added to the vector $\{h^{\ell}-h^{\ell-1}\}$ to obtain the vector $\{h^{\ell}\}$. However, while these are the essential steps of the SIP procedure, several aspects of the method remain to be discussed.

Transfer of Arrays

As noted previously, the coefficient matrix [A] is sparse, with only seven nonzero diagonals. Rather than passing the entire matrix to the SIP Package, only the nonzero diagonals are needed; and because of symmetry of the matrix, only the main diagonal and the three lower diagonals are needed. The three lower diagonals correspond to the conductance arrays CC, CR, and CV. The main diagonal is formed from the three conductance arrays and the array HCOF described in Chapter 2. The right hand side of the matrix equation, $\{q\}$, corresponds to the array RHS described in Chapter 2. The latest estimate of the head distribution $\{h^{\ell-1}\}$, corresponds to the array HNEW. As new estimates of head are calculated by SIP, they are stored in HNEW replacing the previous estimates. Thus input to SIP consists of the following arrays: CC, CR, CV, RHS, HCOF, and HNEW. Output from SIP consists of a new HNEW array. As explained in Chapter 3, the Formulate Procedure is inside the iteration loop; therefore, the input arrays may be modified at each iteration.

Order of Calculation

Experience has shown that if the finite-difference equations are solved in two different orders on alternate iterations, the number of iterations needed to converge to a solution is reduced. The order assumed in the discussion, to this point, has been to begin at the first column, the first row, and the first layer, and to proceed in ascending column order, ascending row order, and ascending layer order. An alternative is to start at the first column, the last row, and the last layer, and to proceed in ascending column order, descending row order, and descending layer order. Using the same ordering of diagonal names used in figure 51, equations similar to equations (97-a...m) can be developed. They are

$$a_{i,j,k} = Z_{i,j,k}/(1+\alpha e_{i,j,k+1}+\beta f_{i,j,k+1}) \tag{107-a}$$

$$b_{i,j,k} = B_{i,j,k}/(1+\gamma e_{i+1,j,k}+\beta g_{i+1,j,k}) \tag{107-b}$$

$$c_{i,j,k} = D_{i,j,k}/(1+\gamma f_{i,j-1,k}+\alpha g_{i,j-1,k}) \tag{107-c}$$

$$A'_{i,j,k} = a_{i,j,k} e_{i,j,k+1} \tag{107-d}$$

$$C'_{i,j,k} = e_{i+1,j,k} b_{i,j,k} \tag{107-e}$$

$$G'_{i,j,k} = f_{i,j-1,k} c_{i,j,k} \tag{107-f}$$

$$R'_{i,j,k} = g_{i,j-1,k} c_{i,j,k} \tag{107-g}$$

$$T'_{i,j,k} = a_{i,j,k} f_{i,j,k+1} \tag{107-h}$$

$$U'_{i,j,k} = b_{i,j,k} g_{i+1,j,k} \tag{107-i}$$

$$\begin{aligned} d_{i,j,k} = {} & E_{i,j,k} + \alpha A'_{i,j,k} + \beta T'_{i,j,k} \\ & + \gamma C'_{i,j,k} + \gamma G'_{i,j,k} + \beta U'_{i,j,k} \\ & + \alpha R'_{i,j,k} - a_{i,j,k} g_{i,j,k+1} - b_{i,j,k} f_{i+1,j,k} \\ & - e_{i,j-1,k} c_{i,j,k} \end{aligned} \tag{107-j}$$

$$e_{i,j,k} = (F_{i,j,k} - \alpha A'_{i,j,k} - \gamma C'_{i,j,k})/d_{i,j,k} \qquad (107\text{-k})$$

$$f_{i,j,k} = (H_{i,j,k} - \beta T'_{i,j,k} - \gamma G'_{i,j,k})/d_{i,j,k} \qquad (107\text{-l})$$

$$g_{i,j,k} = (S_{i,j,k} - \alpha R'_{i,j,k} - \beta U'_{i,j,k})/d_{i,j,k} \qquad (107\text{-m})$$

In the model described herein, equations (107-a...m) and equations (97-a...m) are in effect invoked alternately in successive iterations. The model program actually uses one general set of equations in which the variables are identified by single indices. The ordering of (97) or of (107) is then achieved through the sequence of values assigned to the indices. In the following list of these general equations, the index nll refers to the cell in the previous layer calculated, but in the same row and column as cell n; the indices nrl and ncl are defined analogously. Also, in these equations, the iteration parameters α, β and γ have each been replaced by a single parameter ω as explained in the following section. Note that one additional equation has been added to the list--the equation for v_n, the element of the vector $\{v\}$ corresponding to cell n. This equation can be added inasmuch as v_n can be calculated as soon as the n^{th} rows of the matrices [L] and [U] have been calculated. The equations are

$$a_n = Z_n/(1 + \omega(e_{nll} + f_{nll})) \qquad (108\text{-a})$$

$$b_n = B_n/(1 + \omega(e_{nrl} + g_{nrl})) \qquad (108\text{-b})$$

$$c_n = D_n/(1 + \omega(f_{ncl} + g_{ncl})) \qquad (108\text{-c})$$

$$A'_n = a_n e_{nll} \qquad (108\text{-d})$$

$$C'_n = b_n e_{nrl} \qquad (108\text{-e})$$

$$G'_n = c_n f_{ncl} \qquad (108\text{-f})$$

$$R'_n = c_n g_{ncl} \qquad (108\text{-g})$$

$$T'_n = a_n f_{nll} \qquad (108\text{-h})$$

$$U'_n = b_n g_{nrl} \qquad (108\text{-i})$$

$$d_n = E_n + \omega(A'_n + T'_n + C'_n + G'_n + U'_n + R'_n) - a_n g_{n11} - b_n f_{nrl} - c_n e_{ncl} \quad (108\text{-j})$$

$$e_n = (F_n - \omega(A'_n + C'_n))/d_n \quad (108\text{-k})$$

$$f_n = (H_n - \omega(T'_n + G'_n))/d_n \quad (108\text{-l})$$

$$g_n = (S_n - \omega(R'_n + U'_n))/d_n \quad (108\text{-m})$$

$$v_n = (RES_n - a_n v_{n11} - b_n v_{nrl} - c_n v_{ncl})/d_n \quad (108\text{-n})$$

Since the backward substitution requires all values of e_n, f_n, g_n, and v_n, space is allocated in the SIP Package for four arrays to store those values. Each of these arrays has as many elements as there are cells in the grid.

Iteration Parameters

While Weinstein, Stone and Kwan (1969) define three iteration parameters in their theoretical development, they utilize a single value in practice. Thus the terms α, β and γ of equation (93) are replaced by a single parameter, ω, which multiplies each term on the right side of the equation; however, ω must be cycled through a series of values in successive iterations to achieve satisfactory rates of convergence. In the model described herein, values of ω are calculated from the expression

$$\omega(\lambda) = 1-(WSEED)^{(\lambda-1)/(NPARM-1)} \qquad \lambda=1,2....NPARM \quad (109)$$

where NPARM is the total number of ω values to be used; λ is an index taking on integral values from 1 to NPARM; $w(\lambda)$ is the corresponding iteration parameter value; and WSEED is the iteration parameter "seed", calculated according to rules outlined below, and used as a basis for determining the sequence of ω values.

The value of WSEED is in turn developed as follows. The terms ρ_1, ρ_2, and ρ_3 are calculated for each cell in the mesh using the conductances between that cell and its neighbors, as follows

$$\rho_1 = \frac{CC_{max} + CV_{max}}{CR_{min}} \tag{110}$$

$$\rho_2 = \frac{CR_{max} + CV_{max}}{CC_{min}} \tag{111}$$

$$\rho_3 = \frac{CR_{max} + CC_{max}}{CV_{min}} \tag{112}$$

where CC_{max} for a given cell, i,j,k, is the larger of $CC_{i-1/2,j,k}$ and $CC_{i+1/2,j,k}$, while CC_{min} is the smaller of these values; and similarly CR_{max} is the larger of $CR_{i,j-1/2,k}$ and $CR_{i,j+1/2,k}$, while CR_{min} is the smaller, and CV_{max} is the larger of $CV_{i,j,k-1/2}$ and $CV_{i,j,k+1/2}$, while CV_{min} is the smaller. Using these values, the terms

$$\frac{\pi^2}{2(NCOL)^2(1+\rho_1)} \quad , \quad \frac{\pi 2}{2(NROW)^2(1+\rho_2)} \quad \text{and} \quad \frac{\pi^2}{2(NLAY)^2(1+\rho_3)}$$

are computed for each cell in the grid, where again NCOL is the number of columns in the model, NROW is the number of rows and NLAY is the number of layers; and the minimun value for each of these three terms is taken to be the cell seed. The seed term, WSEED, is then taken as average of all the cell seeds. The iteration parameters, $\omega(\lambda)$, generated from the WSEED value are then used sequentially in successive iterations, recycling each time the entire set has been used (i.e., each NPARM iterations).

The process described above for calculating the sequence of ω values differs slightly from that used by Weinstein, Stone and Kwan (1969), but produces values that are in the same range and that appear to function well in many problems. However, several points should be made regarding iteration parameter selection. First, the process is essentially empirical,

and there is little understanding of why one sequence of parameters performs better than another. Second, the parameters chosen affect the rate of convergence but (assuming that convergence is achieved) should not influence the final solution. Third, the influence of the parameters on the rate of convergence is extremely significant.

The model described herein provides for an additional iteration parameter, referred to here as the acceleration parameter to distinguish it from ω. This parameter, which is designated ACCL in the program, functions as a multiplier of $\{RES^{\ell}\}$; thus where a parameter of this type is not to be used, ACCL is simply assigned a value of one. Parameters similar to ACCL have been used in various versions of SIP (Peaceman, 1977, page 130) although Weinstein, Stone and Kwan (1969) do not employ a parameter of this type. ACCL is not cycled, but rather is assigned a single value by the user. As a general rule, it should initially be given a value of one, and improvement in the rate of convergence should be pursued through adjustment of the seed term, as explained below. If problems with convergence persist, values of ACCL other than one can be tried.

Experience has shown that setting the acceleration/relaxation parameter (ACCL) to 1 and using the seed value calculated by the program does not always produce optimum convergence--that is, the number of iterations required to achieve convergence is not minimum. Convergence rates will deviate from the optimum if the absolute value of head change in each iteration is consistently either too small or too large. When the head change is too large, the computed head overshoots the correct value, and oscillations occur as the head change repeatedly reverses to compensate

for the overshoot. Severe overshoot causes divergence, while moderate overshoot simply slows down convergence. When head change is too small, the opposite problem occurs; head tends to approach the correct value monotonically, but very slowly. In severe situations, the head changes at each iteration may be so small that the criterion for convergence is satisfied at all points, even though the computed heads are still far from the correct values. In such situations, a significant volumetric budget imbalance will occur.

Weinstein, Stone, and Kwan (1969) suggest that a trial and error method can be used to improve the choice of seed. This can be done by making an initial run using the seed calculated by the program or chosen from experience, and using ACCL=1. The trend of head change per iteration, with increasing iteration number, is observed for the iterations of a single time step. There is normally some variation in head change from one iteration to the next due to the cycling of iteration parameters, but this variation is often superimposed on an overall trend in which head change tends either to increase or decrease as iterations continue; it is this overall trend (which is often most evident in the later iterations of the test) that is of interest here. Some oscillations (reversals in sign) of the computed head change are normal during convergence; however, repeated oscillation is a sign of overshoot, indicating that computed head changes are too great for optimal convergence. Head changes which are too small, on the other hand, are indicated by a very flat overall trend. For proper evaluation of the trend, the trial should generally be run for a number of iterations equal to 4 or 5 times the number of iteration parameters, unless convergence occurs before this.

Following the initial trial, the seed is multiplied by a number between two and ten if head changes in the initial trial appear to be too great, and divided by a number between two and ten if those head changes appear to be too small. If the trend in the initial trial is unclear either multiplication or division of the seed may be tried. In any case, a second run is made using the new seed value, and the trend of head change vs. iteration level is again examined. The results are compared with those of the initial trial to see if the rate of convergence has improved. If both runs have converged, the comparison is based on the number of iterations required for convergence; if they have failed to converge, the comparison is based on the head changes observed in the final iterations.

The trial runs can be continued to further refine the choice of seed; in general the seed value will be multiplied or divided by progressively smaller numbers at each step of the procedure. However, it is usually not worthwhile to carry the process too far; multiplication or division of the seed by factors less than 2 is seldom warranted.

In most cases, a satisfactory seed value developed by this procedure will remain satisfactory even though changes in the model are introduced--for example, additional stresses, modifications in boundary conditions or changes in the model mesh. However, if convergence problems arise after such changes, the trial and error procedure can be repeated. It should be noted that the more strongly diagonal the coefficient matrix, the less important the choice of seed will be. Thus, source terms such as evapotranspiration or stream seepage, which affect only coefficients on the main diagonal, normally tend to make the choice of seed less critical; and the addition of such terms to a model seldom necessitates modification of the seed.

For each iteration, the program stores the value of $|\Delta h_\ell|_{max}$, where $|\Delta h_\ell|$ refers to the absolute value of the head change computed during iteration ℓ at a given node, and $|\Delta h_\ell|_{max}$ is the maximum such absolute value for the entire mesh in that iteration. For the last time step of each stress period, a table of $|\Delta h_\ell|_{max}$ values for each iteration of the time step is printed in the program output; at the user's option, a table of these values may be printed for every time step, or for time steps which fall at specified intervals. (See Narrative for Module SIP1AP and Sample Problem Output in Appendix D.) In addition to the $|\Delta h_\ell|_{max}$ value, each entry in the table shows the indices of the node at which the maximum change was recorded, and a sign indicating whether the change was positive or negative. This table can be used in the head change trend evaluations described above, under the assumption that the behavior of the $|\Delta h_\ell|_{max}$ values is representative of the behavior of head change throughout the mesh.

Improvements in the rate of convergence can also be obtained by adjusting the acceleration parameter, ACCL. Increases in ACCL will cause increases in the head change at each iteration, while decreases in ACCL will cause decreases in head change. The trial procedure described above can be used for this case as well; however changes in the seed and in ACCL should never be attempted in the same set of trial runs.

It is sometimes necessary to slow the process of convergence in order to prevent cells from converting to the no-flow condition as a result of head overshoot during an iteration. In these situations, optimal convergence cannot be considered convergence in the minimum rumber of iterations, but rather convergence in the smallest number of iterations that does not involve head overshoot. The procedure of examining head

change per iteration and adjusting iteration parameters can again be used to determine when this condition is being met, and to develop the required seed or ACCL terms.

Strongly Implicit Procedure Package Input

Input to the Strongly Implicit Procedure (SIP) Package is read from the unit specified in IUNIT(9).

FOR EACH SIMULATION

SIP1AL

1.	Data:	MXITER	NPARM
	Format:	I10	I10

SIP1RP

2.	Data:	ACCL	HCLOSE	IPCALC	WSEED	IPRSIP
	Format:	F10.0	F10.0	I10	F10.0	I10

Explanation of Fields Used in Input Instructions

MXITER--is the maximum number of times through the iteration loop in one time step in an attempt to solve the system of finite-difference equations. Fifty iterations are generally sufficient.

NPARM--is the number of iteration parameters to be used. Five parameters are generally sufficient.

ACCL--is the acceleration parameter. It must be greater than zero and is generally equal to one. If a zero is entered, it is changed to one.

HCLOSE--is the head change criterion for convergence. When the maximum absolute value of head change from all nodes during an iteration is less than or equal to HCLOSE, iteration stops.

IPCALC--is a flag indicating where the iteration parameter seed will come from.

0 - the seed will be entered by the user.

1 - the seed will be calculated at the start of the simulation from problem parameters.

WSEED--is the seed for calculating iteration parameters. It is only specified if IPCALC is equal to zero.

IPRSIP--is the printout interval for SIP. If IPRSIP is equal to zero, it is changed to 999. The maximum head change (positive or negative) is printed for each iteration of a time step whenever the time step is an even multiple of IPRSIP. This printout also occurs at the end of each stress period regardless of the value of IPRSIP.

SAMPLE INPUT TO THE SIP PACKAGE(USER SPECIFIES THE SEED)

DATA ITEM	EXPLANATION	INPUT RECORDS				
1	{MXITER, NPARM}_____	50	5			
2	{ACCL, HCLOSE, IPCALC, WSEED, IPRSIP}_____	1.	.01	0	.98	10

SAMPLE INPUT TO THE SIP PACKAGE(PROGRAM CALCULATES THE SEED)

DATA ITEM	EXPLANATION	INPUT RECORDS		
1	{MXITER, NPARM}_____	100	6	
2	{ACCL, HCLOSE, IPCALC, WSEED, IPRSIP}	1.	.01	1

Module Documentation for the Strongly Implicit Procedure Package

The Stongly Implicit Procedure Package (SIP1) consists of three primary modules and two submodules. They are:

Primary Modules

SIP1AL — Allocates space for SIP work arrays.

SIP1RP — Reads control information needed by the SIP Package and calculates iteration parameters if the seed is specified by the user.

SIP1AP — Performs one iteration of the strongly implicit procedure.

Submodules

SSIP1P — Prints the largest head change for each iteration.

SSIP1I — Calculates iteration parameters when the seed is calculated by the program.

Narrative for Module SIP1AL

Module SIP1AL allocates space in the X array for SIP arrays. The four arrays, EL, FL, GL, and V hold intermediate results during the solution process. Each of these contains one element for each model cell. Additionally, three arrays, HDCG, LRCH, and W are required. HDCG holds the maximum head change each iteration and LRCH holds the cell location at which the maximum occurred. The number of elements in HDCG is MXITER, and that in LRCH is three times MXITER, where MXITER is the maximum number of iterations allowed in a time step and is specified by the user. Array W holds iteration parameters. One element in W is used for each of the NPARM iteration parameters. (NPARM is specified by the user.)

Module SIP1AL performs its functions in the following order:

1. Print a message identifying the SIP Package.

2. Read and print MXITER and NPARM.

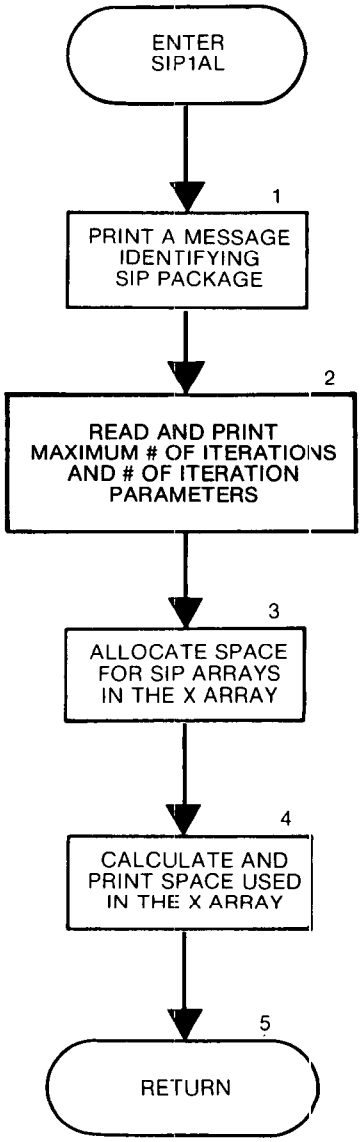
3. Allocate the required space in the X array. The X-array location pointer (ISUM) is saved in variable ISOLD prior to allocation so that the space required for SIP can be calculated in step 4.

4. Calculate and print the space used in the X array. The space used by SIP is ISUM - ISOLD. The total space allocated by all packages so far is ISUM - 1.

5. RETURN.

Flow Chart for Module SIP1AL

X array is the pool of memory space from which space is allocated for arrays used by various packages.



```
      SUBROUTINE SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,LCHDCG,LCLRCH,
     1       LCW,MXITER,NPARM,NCOL,NROW,NLAY,IN,IOUT)
C
C-----VERSION 1110 31DEC1986 SIP1AL
C
C     ******************************************************************
C     ALLOCATE STORAGE IN THE X ARRAY FOR SIP ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------PRINT A MESSAGE IDENTIFYING SIP PACKAGE
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'SIP1 -- STRONGLY IMPLICIT PROCEDURE SOLUTION PACKAGE'
     1,', VERSION 1, 9/1/87',' INPUT READ FROM UNIT',I3)
C
C2------READ AND PRINT MXITER AND NPARM
      READ(IN,2) MXITER,NPARM
    2 FORMAT(2I10)
      WRITE(IOUT,3) MXITER,NPARM
    3 FORMAT(1X,'MAXIMUM OF',I4,' ITERATIONS ALLOWED FOR CLOSURE'/
     1       1X,I2,' ITERATION PARAMETERS')
C
C3------ALLOCATE SPACE FOR THE SIP ARRAYS
      ISOLD=ISUM
      NRC=NROW*NCOL
      ISIZ=NRC*NLAY
      LCEL=ISUM
      ISUM=ISUM+ISIZ
      LCFL=ISUM
      ISUM=ISUM+ISIZ
      LCGL=ISUM
      ISUM=ISUM+ISIZ
      LCV=ISUM
      ISUM=ISUM+ISIZ
      LCHDCG=ISUM
      ISUM=ISUM+MXITER
      LCLRCH=ISUM
      ISUM=ISUM+3*MXITER
      LCW=ISUM
      ISUM=ISUM+NPARM
C
C4------CALCULATE AND PRINT THE SPACE USED IN THE X ARRAY
      ISP=ISUM-ISOLD
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED BY SIP')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module SIP1AL

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.
ISIZ	Module	Number of cells in the grid.
ISOLD	Package	Before this module allocates space, ISOLD is set equal to ISUM. After allocation, ISOLD is subtracted from ISUM to get ISP, the amount of space in the X array allocated by this module.
ISP	Module	Number of words in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM-1.
LCEL	Package	Location in the X array of the first element of array EL.
LCFL	Package	Location in the X array of the first element of array FL.
LCGL	Package	Location in the X array of the first element of array GL.
LCHDCG	Package	Location in the X array of the first element of array HDCG.
LCLRCH	Package	Location in the X array of the first element of array LRCH.
LCV	Package	Location in the X array of the first element of array V.
LCW	Package	Location in the X array of the first element of array W.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MXITER	Package	Maximum number of iterations.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NPARM	Package	Number of iteration parameters.
NRC	Module	Number of cells in a layer.
NROW	Global	Number of rows in the grid.

Narrative for Module SIP1RP

Module SIP1RP reads data for the SIP package: the acceleration parameter (ACCL), the closure criterion (HCLOSE), the iteration-parameter seed (WSEED), a flag indicating whether WSEED is to be calculated or specified by the user (IPCALC), and the interval for printing head change (IPRSIP). If IPCALC is zero, iteration parameters are calculated using WSEED as the seed. Module SIP1RP performs its functions in the following order:

1. Read the data. If ACCL is zero, substitute a default of 1.0. If IPRS1P is less than or equal to zero, substitute an interval of 999 time steps. The defaults are provided as a convenience to the user.

2. Print the data read in step 1.

3. Check IPCALC which is a flag that indicates the source of the iteration-parameter seed (WSEED).

   (a) If IPCALC is not zero, submodule SSIP1I will calculate a seed and the resulting iteration parameters at the start of the first iteration. Print a message telling of this option.

   (b) If IPCALC is zero, use WSEED to calculate iteration parameters. The i-th iteration parameter (I_i) is given by the expression

$$I_i = 1 - (WSEED)^{\frac{i-1}{NPARM-1}}.$$

Print the parameters.

4. RETURN.

Flow Chart for Module SIP1RP

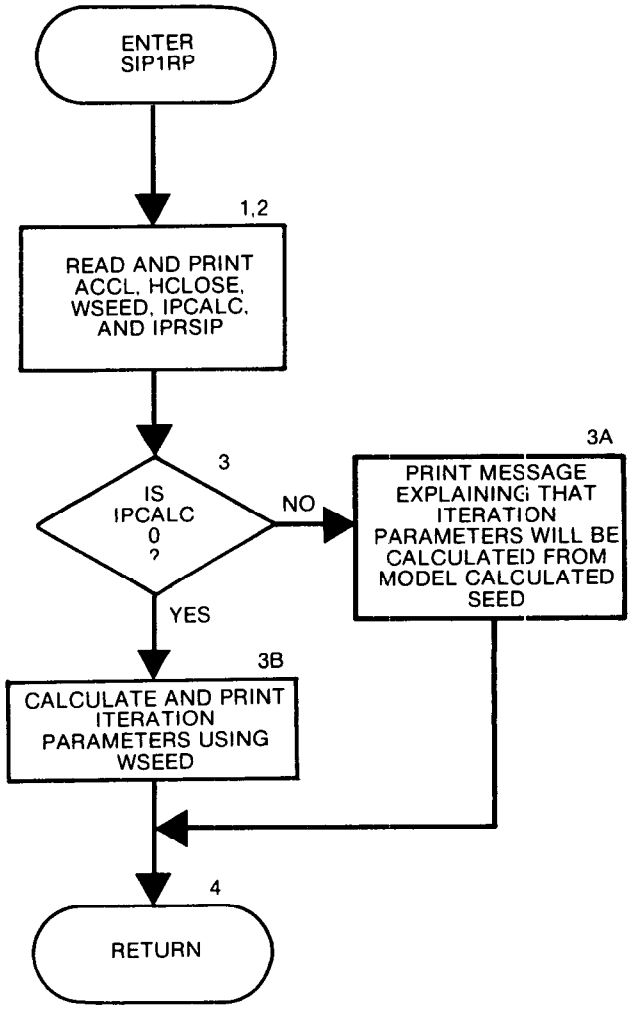
ACCL is a multiplier of calculated head change which is used to control the convergence rate.

HCLOSE is the head change closure criterion. When head change in all model cells is less than or equal to HCLOSE, iteration stops.

WSEED is the seed, specified by the user, on which the calculation of iteration parameters is based if IPCALC is zero.

IPRSIP is the time step interval for printing the maximum head change for each iteration of a time step. Head change is printed every IPRSIP time step. Head change is printed at the end of a stress period regardless of the interval.

IPCALC is a flag. If it is set equal to one, iteration parameters will be calculated from a seed calculated within the program. If it is clear (equal to zero), iteration parameters will be calculated from a seed provided by the user.



```
      SUBROUTINE SIP1RP(NPARM,MXITER,ACCL,HCLOSE,W,IN,IPCALC,IPRSIP,
     1                  IOUT)
C
C-----VERSION 0925 16DEC1982 SIP1RP
C
C     ******************************************************************
C     READ DATA FOR SIP
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION W(NPARM)
C     ------------------------------------------------------------------
C
C1------READ ACCL,HCLOSE,WSEED,IPCALC,IPRSIP
      READ(IN,1) ACCL,HCLOSE,IPCALC,WSEED,IPRSIP
    1 FORMAT(2F10.0,I10,F10.0,I10)
      IF(ACCL.EQ.0.) ACCL=1.
C
C2------PRINT DATA VALUES JUST READ
      WRITE(IOUT,100)
  100 FORMAT(1HO,///57X,'SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE'
     1/57X,43('-'))
      WRITE(IOUT,115) MXITER
  115 FORMAT(1HO,47X,'MAXIMUM ITERATIONS ALLOWED FOR CLOSURE =',I9)
      WRITE(IOUT,120) ACCL
  120 FORMAT(1H ,63X,'ACCELERATION PARAMETER =',G15.5)
      WRITE(IOUT,125) HCLOSE
  125 FORMAT(1H ,52X,'HEAD CHANGE CRITERION FOR CLOSURE =',E15.5)
      IF(IPRSIP.LE.0)IPRSIP=999
      WRITE(IOUT,130) IPRSIP
  130 FORMAT(1H ,52X,'SIP HEAD CHANGE PRINTOUT INTERVAL =',I9)
C
C3------CHECK IF SPECIFIED VALUE OF WSEED SHOULD BE USED OR IF
C3------SEED SHOULD BE CALCULATED
      IF(IPCALC.EQ.0) GO TO 150
C
C3A-----CALCULATE SEED & ITERATION PARAMETERS PRIOR TO 1ST ITERATION
      WRITE(IOUT,140)
  140 FORMAT(1HO,52X,'CALCULATE ITERATION PARAMETERS FROM MODEL',
     1' CALCULATED WSEED')
      GO TO 1000
C
C3B-----USE SPECIFIED VALUE OF WSEED
C3B-----CALCULATE AND PRINT ITERATION PARAMETERS
  150 P1=-1.
      P2=NPARM-1
      DO 160 I=1,NPARM
      P1=P1+1.
  160 W(I)=1.-WSEED**(P1/P2)
      WRITE(IOUT,161) NPARM,WSEED,(W(J),J=1,NPARM)
  161 FORMAT(1HO,/,I5,' ITERATION PARAMETERS CALCULATED FROM',
     1     ' SPECIFIED WSEED =',F11.8,' :'//(10X,6E15.7))
C
C4------RETURN
 1000 RETURN
      END
```

List of Variables for Module SIP1RP

Variable	Range	Definition
ACCL	Package	Acceleration parameter.
HCLOSE	Package	Closure criterion for the iterative procedure.
I	Module	Index for iteration parameters.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPCALC	Package	Flag. ≠ 0, seed for iteration will be calculated in the program. = 0, seed will be specified by the user.
IPRSIP	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
J	Module	Index for iteration parameters.
MXITER	Package	Maximum number of iterations.
NPARM	Package	Number of iteration parameters.
P1	Module	I - 1.
P2	Module	NPARM - 1.
W	Package	DIMENSION (NPARM), Iteration parameters.
WSEED	Module	Seed for calculating iteration parameters.

Narrative for Module SIP1AP

Module SIP1AP performs one iteration of the Strongly Implicit Procedure (SIP) algorithm for solving the flow equation. To save computational time, all arrays are declared one dimensional. The one-dimensional indexes are calculated from the layer, row, and column indexes normally used to access the arrays in three dimensions. Improvement in computational time is achieved because knowledge of the geometry is used to increase computational efficiency, and because calculations are not repeated for identical indices as would be done by internal FORTRAN addressing routines if three-dimensional subscripts were used.

This module is somewhat complex, partly because the SIP solution process requires that the same calculations be performed with two methods of ordering the equations. This is implemented by a generalized algorithm that uses the same computer statements to handle both ordering schemes. Checks are made to detect which ordering scheme is used, and array indexes are calculated accordingly.

Double precision is used for most calculations in this module in order to allow accurate answers to be calculated for a wide range of problems. Mixed precision arithmetic has been avoided by setting double-precision variables equal to single-precision values and then using the double-precision variables to generate fully double-precision expressions, and where necessary by doing the reverse to generate fully single precision expressions.

In the explanations below, no attempt has been made to discuss each use of an assignment statement to change precision, because of the large amount of text this would require. However, when changing this module, care should be used to maintain expressions that have unmixed precision, as mixed precision expressions can cause erroneous results with some compilers.

Module SIP1AP performs its functions in the following order:

1. If the user has specified (IPCALC ≠ 0) that iteration parameters should be calculated by the program, CALL submodule SSIP1I to calculate both the seed and the parameters.

2. Assign values to fields that are constant during an iteration.

3. Initialize the variables that track maximum head change during an iteration.

4. Clear SIP work arrays.

5. Determine the ordering of equations and set the ordering flag (IDIR) accordingly. This flag alternates between 1 and -1 each iteration. Calculate indexes IDNRC and IDNCOL which are used when calculating locations of neighboring cells.

6. Calculate the matrix [U] and intermediate vector {V} using forward substitution. The elements in matrix [L] are used as they are calculated; therefore, they are not saved. In the explanation of SIP concepts, the diagonals in the matrix [U] were designated "e," "f," and "g." The corresponding field names in the program are EL (e lower case), FL (f lower case),

and GL (g lower case). Similarly, the diagonals in the [L] array which are "a," "b," "c," and "d" in the explanation are "AL," "BL," "CL," and "DL" in the program. The codes for the diagonals in matrix [A] in the explanation are the same in the program. The codes for diagonals in [A+B] in the explanation are followed by a "P" in the program. Hence, Z' in the explanation is ZP (Z prime) in the program. The intermediate vector {V} in the explanation is the array "V" in the program.

(a) Set current cell indexes, II, JJ, KK. For normal ordering, the equation order is the same as the order of the loop indexes I,J,K. For reverse ordering, loop indexes I and K are inverted to produce the proper sequence of cells.

(b) Calculate the one-dimensional subscript of the current cell. If this cell is constant head or no flow, skip calculations for this cell and go on to the next.

(c) Calculate the one-dimensional subscripts for the six neighboring cells.

(d) Calculate the one-dimensional subscripts for conductance to each of the six neighboring cells. Since conductances between cells are assigned to array elements at specific cells (for example, CR(I,J,K) stores conductance between cells I,J,K and I,J+1,K), the four or five conductance subscripts are not simply the cell locations of the six neighboring cells as calculated in step 6(c). Also, the subscripts depend on equation ordering.

(e) Calculate or assign variables that are required for forward substitution and involve neighboring cells. Whenever a neighboring cell is outside of the grid, the variables are set to zero.

(1) Neighboring cell is one row back.

(2) Neighboring cell is one row ahead.

(3) Neighboring cell is one column back.

(4) Neighboring cell is one column ahead.

(5) Neighboring cell is one layer back.

(6) Neighboring cell is one layer ahead.

(f) Calculate the components of the upper and lower triangular matrices [U] and [L], which are the factors of matrix [A+B].

(g) Calculate the residual {RES}. The calculation of HNW times HCOF is done in single precision so that the calculation will have precision comparable to similar calculations made in the formulation modules, all of which use single precision.

(h) Calculate the intermediate vector {V}, which is stored in array V. This step completes the forward-substitution process for one cell.

7. Step through the cells solving for head change using back substitution.

(a) Set current cell indexes II, JJ, KK. The ordering is the reverse of that used for forward substitution (step 6(a)).

(b) Calculate the one-dimensional subscript of the current cell. If this cell is constant head or no flow, skip calculations for this cell and go to the next.

(c) Calculate the one-dimensional subscripts for the three neighboring cells behind (relative to the direction of the back-substitution ordering) the current cell.

(d) Back substitute, solving for head change. Store head change in array V in place of the intermediate values of vector {V}. This doubling up of storage is used to save the cost of additional computer storage.

(e) Save the value of head change whose absolute value is largest during this iteration. Also, save the cell location where this head change occurred and the absolute value of the head change.

(f) Add the head change this iteration to head from the previous iteration to get a new estimate of head.

8. Store the head change whose absolute value is greatest this iteration and its cell location in arrays HDCG and LRCH. These may be printed in step 10 at the end of the time step. Set the convergence flag to one if the convergence criterion is met.

9. If the iteration is complete, print the number of iterations for the step; otherwise, RETURN.

10. Print the maximum head change and cell location each iteration if the SIP printout interval (IPRSIP) is reached. Printout occurs at the end of a stress period regardless of the interval.

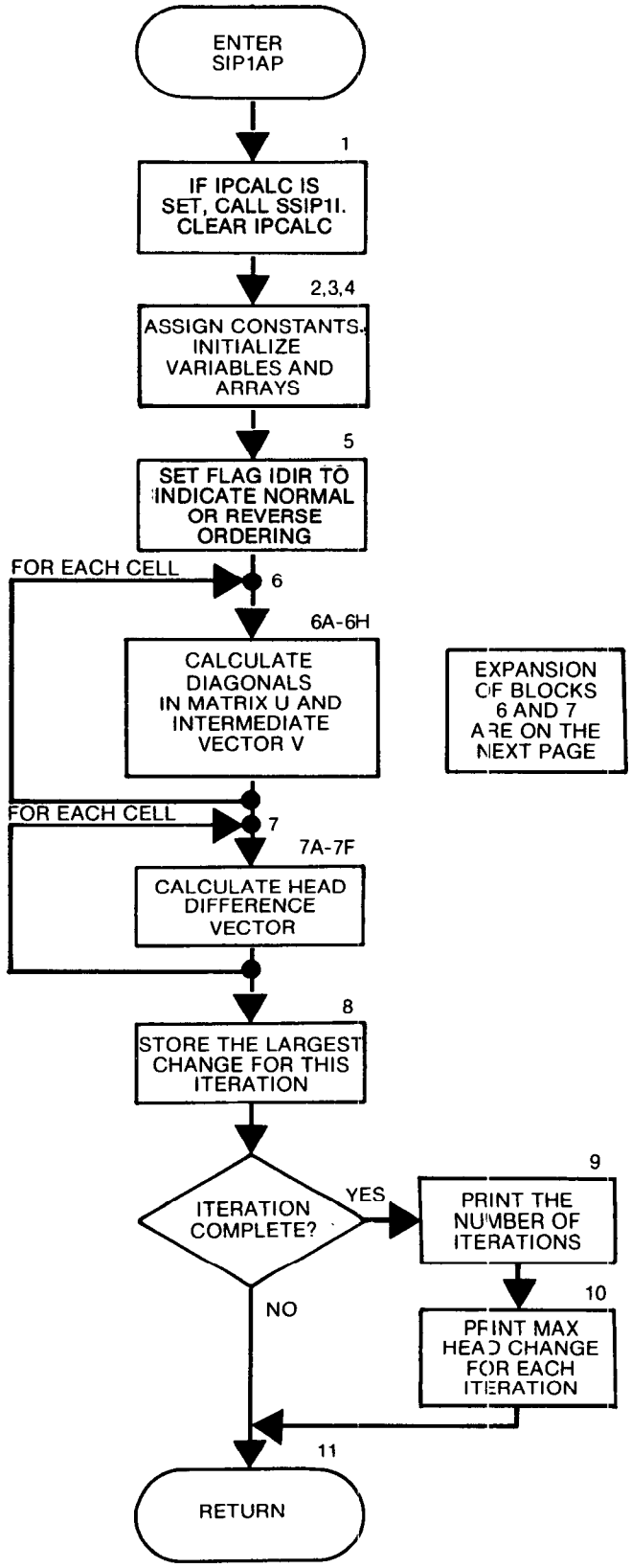
11. RETURN.

Flow Chart for Module SIP1AP

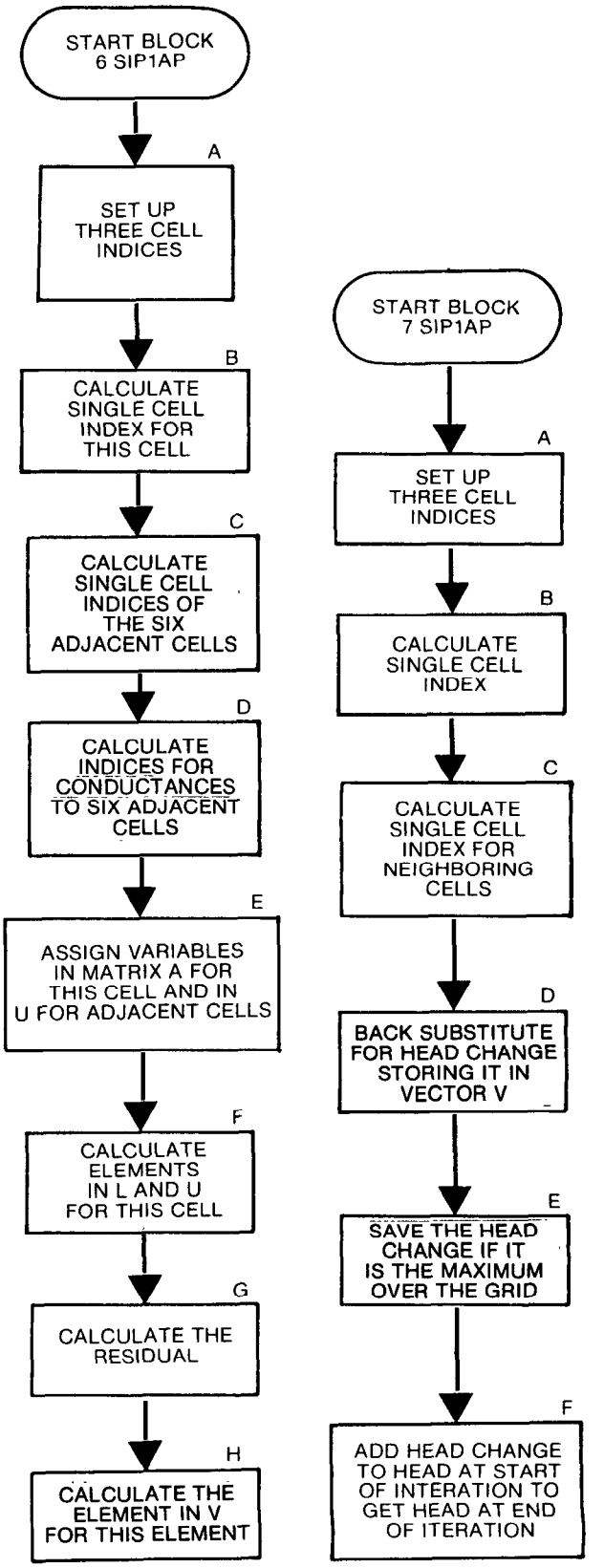
IPCALC is a flag. If it is set equal to one, the program calculates a seed from which iteration parameters are calculated. It may be set by the user at the beginning of the simulation. It is cleared during the first iterations. SSIP1I will never be called more than once. If IPCALC is not set equal to zero, the user specifies the seed for the iteration parameters.

IDIR indicates whether the ordering of equations is normal (1) or reverse (-1).

SSIP1I is a submodule which calculates iteration parameters.



Single Cell Index: In this module, a single index is used to identify each cell. This is in opposition to the three indices (I,J,K) used in most other modules.



```
      SUBROUTINE SIP1AP(HNEW,IBOUND,CR,CC,CV,HCOF,RHS,EL,FL,GL,V,
     1      W,HDCG,LRCH,NPARM,KITER,HCLOSE,ACCL,ICNVG,KSTP,KPER,
     2      IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,IOUT)
C-----VERSION 1656 24JUL1987 SIP1AP
C
C     ******************************************************************
C     SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE -- 1 ITERATION
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,DITPAR,AC,HHCOF,RRHS,XI,DZERO,DONE,RES
      DOUBLE PRECISION Z,B,D,E,F,H,S,AP,TP,CP,GP,UP,RP
      DOUBLE PRECISION ZHNEW,BHNEW,DHNEW,FHNEW,HHNEW,SHNEW
      DOUBLE PRECISION AL,BL,CL,DL,ELNCL,FLNCL,GLNCL
      DOUBLE PRECISION ELNRL,FLNRL,GLNRL,ELNLL,FLNLL,GLNLL
      DOUBLE PRECISION VNRL,VNCL,VNLL,ELXI,FLXI,GLXI,VN,HCFHNW
C
      DIMENSION HNEW(NODES), IBOUND(NODES), CR(NODES), CC(NODES),
     1  CV(NODES), HCOF(NODES), RHS(NODES), EL(NODES), FL(NODES),
     2  GL(NODES), V(NODES), W(NPARM), HDCG(MXITER), LRCH(3,MXITER)
C     ------------------------------------------------------------------
C
C1------CALCULATE ITERATION PARAMETERS IF FLAG IS SET.  THEN
C1------CLEAR THE FLAG SO THAT CALCULATION IS DONE ONLY ONCE.
      IF(IPCALC.NE.0)
     1     CALL SSIP1I(CR,CC,CV,IBOUND,NPARM,W,NCOL,NROW,NLAY,IOUT)
      IPCALC=0
C
C2------ASSIGN VALUES TO FIELDS THAT ARE CONSTANT DURING AN ITERATION
      DZERO=0.
      DONE=1.
      AC=ACCL
      NRC=NROW*NCOL
      NTH=MOD(KITER-1,NPARM)+1
      DITPAR=W(NTH)
C
C3------INITIALIZE VARIABLE THAT TRACKS MAXIMUM HEAD CHANGE DURING
C3------THE ITERATION
      BIGG=0.
C
C4------CLEAR SIP WORK ARRAYS.
      DO 100 I=1,NODES
      EL(I)=0.
      FL(I)=0.
      GL(I)=0.
  100 V(I)=0.
C
C5------SET NORMAL/REVERSE EQUATION ORDERING FLAG (1 OR -1) AND
C5------CALCULATE INDEXES DEPENDENT ON ORDERING
      IDIR=1
      IF(MOD(KITER,2).EQ.0)IDIR=-1
      IDNRC=IDIR*NRC
      IDNCOL=IDIR*NCOL
C
C6------STEP THROUGH CELLS CALCULATING INTERMEDIATE VECTOR V
C6------USING FORWARD SUBSTITUTION
      DO 150 K=1,NLAY
      DO 150 I=1,NROW
      DO 150 J=1,NCOL
C
C6A-----SET UP CURRENT CELL LOCATION INDEXES.  THESE ARE DEPENDENT
C6A-----ON THE DIRECTION OF EQUATION ORDERING.
      IF(IDIR.LE.0)GO TO 120
      II=I
      JJ=J
      KK=K
      GO TO 122
  120 II=NROW-I+1
      JJ=J
      KK=NLAY-K+1
```

```
C
C6B-----CALCULATE 1 DIMENSIONAL SUBSCRIPT OF CURRENT CELL AND
C6B-----SKIP CALCULATIONS IF CELL IS NOFLOW OR CONSTANT HEAD
  122 N=JJ+(II-1)*NCOL+(KK-1)*NRC
      IF(IBOUND(N).LE.0)GO TO 150
C
C6C-----CALCULATE 1 DIMENSIONAL SUBSCRIPTS FOR LOCATING THE 6
C6C-----SURROUNDING CELLS
      NRN=N+IDNCOL
      NRL=N-IDNCOL
      NCN=N+1
      NCL=N-1
      NLN=N+IDNRC
      NLL=N-IDNRC
C
C6D-----CALCULATE 1 DIMENSIONAL SUBSCRIPTS FOR CONDUCTANCE TO THE 6
C6D-----SURROUNDING CELLS.  THESE DEPEND ON ORDERING OF EQUATIONS.
      IF(IDIR.LE.0)GO TO 124
      NCF=N
      NCD=NCL
      NRB=NRL
      NRH=N
      NLS=N
      NLZ=NLL
      GO TO 126
  124 NCF=N
      NCD=NCL
      NRB=N
      NRH=NRN
      NLS=NLN
      NLZ=N
C
C6E-----ASSIGN VARIABLES IN MATRICES A & U INVOLVING ADJACENT CELLS
C6E1----NEIGHBOR IS 1 ROW BACK
  126 B=DZERO
      ELNRL=DZERO
      FLNRL=DZERO
      GLNRL=DZERO
      BHNEW=DZERO
      VNRL=DZERO
      IF(I.EQ.1) GO TO 128
      B=CC(NRB)
      ELNRL=EL(NRL)
      FLNRL=FL(NRL)
      GLNRL=GL(NRL)
      BHNEW=B*HNEW(NRL)
      VNRL=V(NRL)
C
C6E2----NEIGHBOR IS 1 ROW AHEAD
  128 H=DZERO
      HHNEW=DZERO
      IF(I.EQ.NROW) GO TO 130
      H=CC(NRH)
      HHNEW=H*HNEW(NRN)
C
C6E3----NEIGHBOR IS 1 COLUMN BACK
  130 D=DZERO
      ELNCL=DZERO
      FLNCL=DZERO
      GLNCL=DZERO
      DHNEW=DZERO
      VNCL=DZERO
      IF(J.EQ.1) GO TO 132
      D=CR(NCD)
      ELNCL=EL(NCL)
      FLNCL=FL(NCL)
      GLNCL=GL(NCL)
      DHNEW=D*HNEW(NCL)
      VNCL=V(NCL)
```

```
C
C6E4----NEIGHBOR IS 1 COLUMN AHEAD
  132 F=DZERO
      FHNEW=DZERO
      IF(J.EQ.NCOL) GO TO 134
      F=CR(NCF)
      FHNEW=F*HNEW(NCN)
C
C6E5----NEIGHBOR IS 1 LAYER BEHIND
  134 Z=DZERO
      ELNLL=DZERO
      FLNLL=DZERO
      GLNLL=DZERO
      ZHNEW=DZERO
      VNLL=DZERO
      IF(K.EQ.1) GO TO 136
      Z=CV(NLZ)
      ELNLL=EL(NLL)
      FLNLL=FL(NLL)
      GLNLL=GL(NLL)
      ZHNEW=Z*HNEW(NLL)
      VNLL=V(NLL)
C
C6E6----NEIGHBOR IS 1 LAYER AHEAD
  136 S=DZERO
      SHNEW=DZERO
      IF(K.EQ.NLAY) GO TO 138
      S=CV(NLS)
      SHNEW=S*HNEW(NLN)
C
C6E7----CALCULATE THE NEGATIVE SUM OF ALL CONDUCTANCES TO NEIGHBORING
C6E7----CELLS
  138 E=-Z-B-D-F-H-S
C
C6F-----CALCULATE COMPONENTS OF THE UPPER AND LOWER MATRICES, WHICH
C6F-----ARE THE FACTORS OF MATRIX (A+B)
      AL=Z/(DONE+DITPAR*(ELNLL+FLNLL))
      BL=B/(DONE+DITPAR*(ELNRL+GLNRL))
      CL=D/(DONE+DITPAR*(FLNCL+GLNCL))
      AP=AL*ELNLL
      CP=BL*ELNRL
      GP=CL*FLNCL
      RP=CL*GLNCL
      TP=AL*FLNLL
      UP=BL*GLNRL
      HHCOF=HCOF(N)
      DL=E+HHCOF+DITPAR*(AP+TP+CP+GP+UP+RP)-AL*GLNLL-BL*FLNRL-CL*ELNCL
      EL(N)=(F-DITPAR*(AP+CP))/DL
      FL(N)=(H-DITPAR*(TP+GP))/DL
      GL(N)=(S-DITPAR*(RP+UP))/DL
C
C6G-----CALCULATE THE RESIDUAL
      RRHS=RHS(N)
      HNW=HNEW(N)
      HCFHNW=HNW*HCOF(N)
      RES=RRHS-ZHNEW-BHNEW-DHNEW-E*HNEW(N)-HCFHNW-FHNEW-HHNEW-SHNEW
C
C6H-----CALCULATE THE INTERMEDIATE VECTOR V
      V(N)=(AC*RES-AL*VNLL-BL*VNRL-CL*VNCL)/DL
C
  150 CONTINUE
C
C7------STEP THROUGH EACH CELL AND SOLVE FOR HEAD CHANGE BY BACK
C7------SUBSTITUTION
      DO 160 K=1,NLAY
      DO 160 I=1,NROW
      DO 160 J=1,NCOL
C
```

```
C7A-----SET UP CURRENT CELL LOCATION INDEXES.  THESE ARE DEPENDENT
C7A-----ON THE DIRECTION OF EQUATION ORDERING.
      IF(IDIR.LT.0) GO TO 152
      KK=NLAY-K+1
      II=NROW-I+1
      JJ=NCOL-J+1
      GO TO 154
  152 KK=K
      II=I
      JJ=NCOL-J+1
C
C7B-----CALCULATE 1 DIMENSIONAL SUBSCRIPT OF CURRENT CELL AND
C7B-----SKIP CALCULATIONS IF CELL IS NOFLOW OR CONSTANT HEAD
  154 N=JJ+(II-1)*NCOL+(KK-1)*NRC
      IF(IBOUND(N).LE.0)GO TO 160
C
C7C-----CALCULATE 1 DIMENSIONAL SUBSCRIPTS FOR THE 3 NEIGHBORING CELLS
C7C-----BEHIND (RELATIVE TO THE DIRECTION OF THE BACK SUBSTITUTION
C7C-----ORDERING) THE CURRRENT CELL.
      NC=N+1
      NR=N+IDNCOL
      NL=N+IDNRC
C
C7D-----BACK SUBSTITUTE, STORING HEAD CHANGE IN ARRAY V IN PLACE OF
C7D-----INTERMEDIATE FORWARD SUBSTITUTION VALUES.
      ELXI=DZERO
      FLXI=DZERO
      GLXI=DZERO
      IF(JJ.NE.NCOL) ELXI=EL(N)*V(NC)
      IF(I.NE.1) FLXI=FL(N)*V(NR)
      IF(K.NE.1) GLXI=GL(N)*V(NL)
      VN=V(N)
      V(N)=VN-ELXI-FLXI-GLXI
C
C7E-----GET THE ABSOLUTE HEAD CHANGE. IF IT IS MAX OVER GRID SO FAR.
C7E-----THEN SAVE IT ALONG WITH CELL INDICES AND HEAD CHANGE.
      TCHK=ABS(V(N))
      IF (TCHK.LE.BIGG) GO TO 155
      BIGG=TCHK
      BIG=V(N)
      IB=II
      JB=JJ
      KB=KK
C
C7F-----ADD HEAD CHANGE THIS ITERATION TO HEAD FROM THE PREVIOUS
C7F-----ITERATION TO GET A NEW ESTIMATE OF HEAD.
  155 XI=V(N)
      HNEW(N)=HNEW(N)+XI
C
  160 CONTINUE
C
C8------STORE THE LARGEST ABSOLUTE HEAD CHANGE (THIS ITERATION) AND
C8------AND ITS LOCATION.
      HDCG(KITER)=BIG
      LRCH(1,KITER)=KB
      LRCH(2,KITER)=IB
      LRCH(3,KITER)=JB
      ICNVG=0
      IF(BIGG.LE.HCLOSE) ICNVG=1
C
C9------IF END OF TIME STEP, PRINT # OF ITERATIONS THIS STEP
      IF(ICNVG.EQ.0 .AND. KITER.NE.MXITER) GO TO 600
      IF(KSTP.EQ.1) WRITE(IOUT,500)
  500 FORMAT(1HO)
      WRITE(IOUT,501) KITER,KSTP,KPER
  501 FORMAT(1X,I5,' ITERATIONS FOR TIME STEP',I4,' IN STRESS PERIOD',
     1        I3)
C
C10-----PRINT HEAD CHANGE EACH ITERATION IF PRINTOUT INTERVAL IS REACHED
      IF(ICNVG.EQ.0 .OR. KSTP.EQ.NSTP .OR. MOD(KSTP,IPRSIP).EQ.0)
     1      CALL SSIP1P(HDCG,LRCH,KITER,MXITER,IOUT)
C
C11-----RETURN
600   RETURN
C
      END
```

List of Variables for Module SIP1AP

Variable	Range	Definition
AC	Module	Double-precision acceleration parameter (ACCL).
ACCL	Package	Acceleration parameter.
AL	Module	Diagonal from the lower factor. (AL stands for A-lower case.)
AP	Module	Diagonal element in the modified coefficient matrix. (AP stands for A-prime.)
B	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the last row.
BHNEW	Module	Head in the adjacent cell which is in the last row.
BIG	Module	Largest head change for an iteration.
BIGG	Module	Largest absolute value of head change for an iteration.
BL	Module	Diagonal from the lower factor. (BL stands for B-lower case.)
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CL	Module	Diagonal from the lower factor. (CL stands for C-lower case.)
CP	Module	Diagonal element in the modified coefficient matrix. (CP stands for C-prime.)
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K)
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
D	Module	Diagonal label in the coefficient matrix. Conductance from the adjacent node which is in the last column.
DHNEW	Module	Head in the adjacent cell which is in the last column.
DITPAR	Module	Double-precision iteration parameter.
DL	Module	Diagonal from the lower factor. (DL stands for D-lower case.)
DONE	Module	Double-precision field containing a one.
DZERO	Module	Double-precision field containing a zero.
E	Module	Main diagonal in the coefficient matrix.
EL	Module	DIMENSION (NODES), Diagonal from the upper factor. (EL stands for E-lower case.)
ELNCL	Module	EL (E-lower case) from the cell in the last column.
ELNLL	Module	EL (E-lower case) from the cell in the last layer.
ELNRL	Module	EL (E-lower case) from the cell in the last row.
ELXI	Module	Intermediate result.
F	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the next column.
FHNEW	Module	Head in the adjacent cell which is in the next column.
FL	Module	DIMENSION (NODES), Diagonal from the upper factor. (FL stands for F-lower case.)
FLNCL	Module	FL (F-lower case) from the cell in the last column.
FLNLL	Module	FL (F-lower case) from the cell in the last layer.
FLNRL	Module	FL (F-lower case) from the cell in the last row.

List of Variables for Module SIP1AP (Continued)

Variable	Range	Definition
FLXI	Module	Intermediate result.
GL	Module	DIMENSION (NODES), Diagonal from the upper factor. (GL stands for G-lower case.)
GLNCL	Module	GL (G-lower case) from the cell in the last column.
GLNLL	Module	GL (G-lower case) from the cell in the last layer.
GLNRL	Module	GL (G-lower case) from the cell in the last row.
GLXI	Module	Intermediate result.
GP	Module	Diagonal element in the modified coefficient matrix. (GP stands for G-prime.)
H	Module	Diagonal label in the coefficient matrix. Conductance from the adjacent node which is in the next row.
HCFHNW	Module	Product of head and HCOF for a cell.
HCLOSE	Package	Closure criterion for the iterative procedure.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HDCG	Package	DIMENSION (MXITER), Maximum head change for each iteration.
HHCOF	Module	Double-precision HCOF.
HHNEW	Module	Head in the adjacent cell which is in the next row.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
HNW	Module	Temporary field for HNEW(N).
I	Module	Index for nodes and rows.
IB	Module	Row number of the cell having the largest head change.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
ICNVG	Global	Flag is set equal to one when the iteration procedure has converged.
IDIR	Module	Indicator for direction of solution algorithm. +1 - forward -1 - reverse
IDNCOL	Module	Intermediate result used to calculate indices.
IDNRC	Module	Intermediate result used to calculate indices.
II	Module	Row number.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPCALC	Package	Flag. = 0, iteration parameter seed (WSEED) is entered by the user. = 1, seed is calculated in the program.
IPRSIP	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
J	Module	Index for columns.
JB	Module	Column number of the cell having the largest head change.
JJ	Module	Column index.
K	Module	Index for layers.
KB	Module	Layer of the cell having the largest head change.
KITER	Global	Iteration counter. Reset at the start of each time step.

List of Variables for Module SIP1AP (Continued)

Variable	Range	Definition
KK	Module	Layer index.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
LRCH	Package	DIMENSION (MXITER), Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MXITER	Package	Maximum number of iterations.
N	Module	Cell index.
NC	Module	Index for the adjacent cell in the last column.
NCD	Module	One-dimensional subscript of conductance to the adjacent cell which is in the last column.
NCF	Module	One-dimensional subscript of conductance to the adjacent cell which is in the next column.
NCL	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the last column.
NCN	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the next column.
NCOL	Global	Number of columns in the grid.
NL	Module	Index for the adjacent cell in the last layer.
NLAY	Global	Number of layers in the grid.
NLL	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the last layer.
NLN	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the next layer.
NLS	Module	One-dimensional subscript of conductance to the adjacent cell which is in the next layer.
NLZ	Module	One-dimensional subscript of conductance to the adjacent cell which is in the last layer.
NODES	Global	Number of cells (nodes) in the finite-difference grid.
NPARM	Package	Number of iteration parameters.
NR	Module	Index for the adjacent cell in the last row.
NRB	Module	One-dimensional subscript of conductance to the adjacent cell which is in the last row.
NRC	Module	Number of cells in the layer.
NRH	Module	One-dimensional subscript of conductance to the adjacent cell which is in the next row.
NRL	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the last row.
NRN	Module	One-dimensional subscript of the cell index of the adjacent cell which is in the next row.
NROW	Global	Number of rows in the grid.
NSTP	Global	Number of time steps in the current stress period.
NTH	Module	Index for iteration parameters.
RES	Module	Residual.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.

List of Variables for Module SIP1AP (Continued)

Variable	Range	Definition
RP	Module	Diagonal element in the modified coefficient matrix. (RP stands for R-prime.)
RRHS	Module	Double-precision right hand side of the equation.
S	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the next layer.
SHNEW	Module	Head in the adjacent cell which is in the next layer.
TCHK	Module	Absolute value of head change for a single cell.
TP	Module	Diagonal element in the modified coefficient matrix. (TP stands for T-prime.)
UP	Module	Diagonal element in the modified coefficient matrix. (UP stands for U-prime.)
V	Package	DIMENSION (NODES), Intermediate result.
VN	Module	Temporary double-precision V(N).
VNCL	Module	Element in the intermediate vector for the cell in the last column.
VNLL	Module	Element in the intermediate vector for the cell in the last layer.
VNRL	Module	Element in the intermediate vector for the cell in the last row.
W	Package	DIMENSION (NPARM), Iteration parameters.
XI	Module	Double-precision V(N).
Z	Module	Diagonal label in the coefficient matrix--conductance from the adjacent node which is in the last layer.
ZHNEW	Module	Head in the adjacent cell which is in the last layer.

Narrative for Module SSIP1P

Submodule SSIP1P prints the largest value of head change (HDCG) out of all cells for each iteration of a time step. Also printed is the cell location (LRCH) where the change occurs. The submodule is so short that no numbered comments are used and no flow chart is provided.

```
      SUBROUTINE SSIP1P(HDCG,LRCH,KITER,MXITER,IOUT)
C
C
C-----VERSION 1636 24JUL1987 SSIP1P
C     ******************************************************************
C     PRINT MAXIMUM HEAD CHANGE FOR EACH ITERATION DURING A TIME STEP
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DIMENSION HDCG(MXITER), LRCH(3,MXITER)
C     ------------------------------------------------------------------
C
      WRITE(IOUT,5)
    5 FORMAT(1HO,'MAXIMUM HEAD CHANGE FOR EACH ITERATION:'/
     1    1HO,5(' HEAD CHANGE LAYER,ROW,COL')/1X,132('-'))
      WRITE (IOUT,10) (HDCG(J),(LRCH(I,J),I=1,3),J=1,KITER)
   10 FORMAT((1X,5(G12.4,' (',I3,',',I3,',',I3,')')))
      WRITE(IOUT,11)
   11 FORMAT(1HO)
C
      RETURN
C
      END
```

List of Variables for Module SSIP1P

Variable	Range	Definition
HDCG	Package	DIMENSION (MXITER), Maximum head change for each iteration.
I	Module	Index for cell location.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
J	Module	Index for iterations.
KITER	Global	Iteration counter. Reset at the start of each time step.
LRCH	Package	DIMENSION (MXITER), Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MXITER	Package	Maximum number of iterations.

Narrative for Module SSIP1I

Submodule SSIP1I calculates an iteration-parameter seed using model-conductance values and grid dimensions. Although a single seed is required, the method of calculation requires that three-directional seeds be calculated for each active cell. Then a cell seed, the minimum of the three, is selected. Finally, all the cell seeds are averaged to give the grid seed. This grid seed is then used to calculate the iteration parameters. The minimum cell seed is also printed.

Submodule SSIP1I performs its functions in the following order:

1. Calculate constants and initialize variables. In order to calculate the average cell seed, accumulators AVGSUM (sum of the cell seeds) and NODES (sum of the active cells for which a seed is calculated) are required. These are initialized to zero. WMINMN is used to store the smallest cell seed. Since this value must always be less than one, it is initialized to 1.0. The three coefficients, CCOL, CROW, and CLAY are set equal to $\pi^2/2(NCOL)^2$, $\pi^2/2(NROW)^2$, and $\pi^2/2(NLAY)^2$, respectively.

2. Loop through all cells, calculating a cell seed for each active cell.

   (a) Find the conductances from the cell to each of the six adjacent cells. Conductance across the grid boundary is set equal to zero.

   (b) Find the maximum and minimum of the two conductances in the row direction (DFMX, DFMN), in the column direction (BHMX, BHMN), and in the vertical direction (ZSMX, ZSMN). If the minimum is zero (which indicates that a neighbor is no flow), set the minimum equal to the maximum.

(c) Calculate three-directional seeds (WCOL, WROW, WLAY) using the relations

```
WCOL = CCOL/(1. + (BHMX + ZSMX)/DFMN);
WROW = CROW/(1. + (DFMX + ZSMX)/BHMN); and
WLAY = CLAY/(1. + (DFMX + BHMX)/ZSMN).
```

If the minimum conductance is zero (that is, both the minimum and the maximum are zero), set the seed equal to 1.0. This value will be ignored when the cell seed (the minimum-directional seed) is selected in step 2(d) because any valid seed will be less than 1.0.

(d) Select the minimum of the three-directional seeds as the cell seed. If it is the smallest cell seed used so far, store it in WMINMN. Accumulate the sum of the cell seeds and the total number of active cells so that the average of all cell seeds can be calculated in step 3.

3. Calculate the grid seed (the average cell seed) and print it along with the minimum seed.

4. Calculate and print iteration parameters using the grid seed with the relation

$$I_i = 1 - (SEED)^{\frac{i-1}{NPARM - 1}}$$

where

I_i is the i-th iteration parameter, and

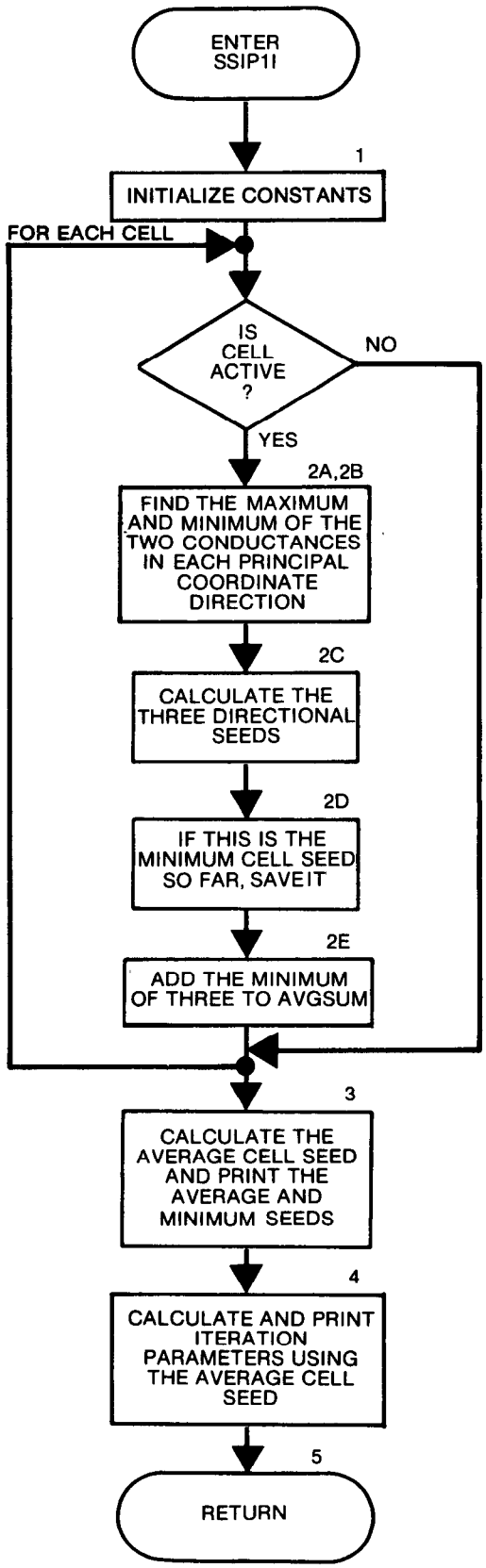
NPARM is the number of iteration parameters.

5. RETURN.

Flow Chart for Module SSIP1I

Seed: the "grid seed" is the single parameter used to calculate the iteration parameters. To calculate the grid seed, several intermediate variables, called "cell seeds," are used. For each cell, three "directional seeds" are calculated. The minimum directional seed for a cell is the "cell seed." The "grid seed" is the average of the cell seeds.

AVGSUM is an accumulator to which each cell seed is added. It is then divided by the number of cells to obtain the average cell seed which is used as the grid seed.



```
      SUBROUTINE SSIP1I(CR,CC,CV,IBOUND,NPARM,W,NCOL,NROW,NLAY,
     1           IOUT)
C
C-----VERSION 1417 12MAY1987 SSIP1I
C     ******************************************************************
C     CALCULATE AN ITERATION PARAMETER SEED AND USE IT TO CALCULATE SIP
C     ITERATION PARAMETERS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION CR(NCOL,NROW,NLAY),CC(NCOL,NROW,NLAY)
     1       ,CV(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),W(NPARM)
C
      DOUBLE PRECISION DWMIN,AVGSUM
C     ------------------------------------------------------------------
C
C1------CALCULATE CONSTANTS AND INITIALIZE VARIABLES
      PIEPIE=9.869604
      R=NROW
      C=NCOL
      ZL=NLAY
      CCOL=PIEPIE/(2.*C*C)
      CROW=PIEPIE/(2.*R*R)
      CLAY=PIEPIE/(2.*ZL*ZL)
      WMINMN=1.
      AVGSUM=0.
      NODES=0
C
C2------LOOP THROUGH ALL CELLS, CALCULATING A SEED FOR EACH CELL
C2------THAT IS ACTIVE
      DO 100 K=1,NLAY
      DO 100 I=1,NROW
      DO 100 J=1,NCOL
      IF(IBOUND(J,I,K).LE.0) GO TO 100
C
C2A-----CONDUCTANCE FROM THIS CELL
C2A-----TO EACH OF THE 6 ADJACENT CELLS
      D=0.
      IF(J.NE.1) D=CR(J-1,I,K)
      F=0.
      IF(J.NE.NCOL) F=CR(J,I,K)
      B=0.
      IF(I.NE.1) B=CC(J,I-1,K)
      H=0.
      IF(I.NE.NROW) H=CC(J,I,K)
      Z=0.
      IF(K.NE.1) Z=CV(J,I,K-1)
      S=0.
      IF(K.NE.NLAY) S=CV(J,I,K)
C
C2B-----FIND THE MAXIMUM AND MINIMUM OF THE 2 CONDUCTANCE COEFFICIENTS
C2B-----IN EACH PRINCIPAL COORDINATE DIRECTION
```

```
      DFMX=AMAX1(D,F)
      BHMX=AMAX1(B,H)
      ZSMX=AMAX1(Z,S)
      DFMN=AMIN1(D,F)
      BHMN=AMIN1(B,H)
      ZSMN=AMIN1(Z,S)
      IF(DFMN.EQ.0.) DFMN=DFMX
      IF(BHMN.EQ.0.) BHMN=BHMX
      IF(ZSMN.EQ.0.) ZSMN=ZSMX
C
C2C-----CALCULATE A SEED IN EACH PRINCIPAL COORDINATE DIRECTION
      WCOL=1.
      IF(DFMN.NE.0.) WCOL=CCOL/(1.+(BHMX+ZSMX)/DFMN)
      WROW=1.
      IF(BHMN.NE.0.) WROW=CROW/(1.+(DFMX+ZSMX)/BHMN)
      WLAY=1.
      IF(ZSMN.NE.0.) WLAY=CLAY/(1.+(DFMX+BHMX)/ZSMN)
C
C2D-----SELECT THE CELL SEED, WHICH IS THE MINIMUM SEED OF THE 3.
C2D-----SELECT THE MINIMUM SEED OVER THE WHOLE GRID.
      WMIN=AMIN1(WCOL,WROW,WLAY)
      WMINMN=AMIN1(WMINMN,WMIN)
C
C2E-----ADD THE CELL SEED TO THE ACCUMULATOR AVGSUM FOR USE
C2E-----IN GETTING THE AVERAGE SEED.
      DWMIN=WMIN
      AVGSUM=AVGSUM+DWMIN
      NODES=NODES+1
C
  100 CONTINUE
C
C3------CALCULATE THE AVERAGE SEED OF THE CELL SEEDS, AND PRINT
C3------THE AVERAGE AND MINIMUM SEEDS.
      TMP=NODES
      AVGMIN=AVGSUM
      AVGMIN=AVGMIN/TMP
      WRITE(IOUT,101) AVGMIN,WMINMN
  101 FORMAT(1H0,'AVERAGE SEED =',F11.8/1X,'MINIMUM SEED =',F11.8)
C
C4------CALCULATE AND PRINT ITERATION PARAMETERS FROM THE AVERAGE SEED
      P1=-1.
      P2=NPARM-1
      DO 50 I=1,NPARM
      P1=P1+1.
   50 W(I)=1.-AVGMIN**(P1/P2)
      WRITE(IOUT,150) NPARM,(W(J),J=1,NPARM)
  150 FORMAT(1H0,/,I5,' ITERATION PARAMETERS CALCULATED FROM',
     1       ' AVERAGE SEED:'//(10X,6E15.7))
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module SSIP1I

Variable	Range	Definition
AVGMIN	Module	Mean WMIN.
AVGSUM	Module	Sum of all of WMIN's.
B	Module	Conductance between this node and the one to the rear.
BHMN	Module	Minimum of B and H (if the minimum is 0, it is the maximum).
BHMX	Module	Maximum of B and H.
C	Module	Number of columns.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CCOL	Module	Intermediate factor.
CLAY	Module	Intermediate factor.
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CROW	Module	Intermediate factor.
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
D	Module	Conductance between this node and the one to the left.
DFMN	Module	Minimum of D and F (if the minimum is 0, it is the maximum).
DFMX	Module	Maximum of D and F.
DWMIN	Module	Double precision WMIN.
F	Module	Conductance between this node and the one to the right.
H	Module	Conductance between this node and the one to the front.
I	Module	Index for rows.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell

List of Variables for Module SSIP1I (Continued)

Variable	Range	Definition
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
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.
NODES	Module	Number of variable-head (IBOUND > 0) cells in the grid.
NPARM	Package	Number of iteration parameters.
NROW	Global	Number of rows in the grid.
PIEPIE	Module	PI squared.
P1	Module	Index for the number of parameters.
P2	Module	NPARM-1.
R	Module	Number of rows.
S	Module	Conductance between this node and the one below.
TMP	Module	Temporary field for nodes.
W	Package	DIMENSION (NPARM), Iteration parameters.
WCOL	Module	Seed in the column direction for a cell.
WLAY	Module	Seed in the layer direction for a cell.
WMIN	Module	Minimum of (WCOL, WLAY, WROW).
WMINMN	Module	Minimum WMIN.
WROW	Module	Seed in the row direction for a cell.
Z	Module	Conductance between this node and the one above.
ZL	Module	Number of layers.
ZSMN	Module	Minimum of Z and S (if minimum is 0, it is the maximum).
ZSMX	Module	Maximum of Z and S.

CHAPTER 13

SLICE-SUCCESSIVE OVERRELAXATION PACKAGE

Conceptualization and Implementation

Successive overrelaxation is another method for solving large systems of linear equations by means of iteration. It is implemented in the model discussed herein through the Slice Successive Overrelaxation (SSOR) Package. Background material on the successive overrelaxation approach can be found in many standard references, including those already noted by Peaceman (1977), Crichlow (1977) and Remson, Hornberger and Molz (1971).

The successive overrelaxation technique is implemented in the SSOR Package by dividing the finite difference grid into vertical "slices," as shown in figure 54, and grouping the node equations into discrete sets, each set corresponding to a slice. In every iteration, these sets of equations are processed in turn, resulting in a new set of estimated head values for each slice. As the equations for each slice are processed, they are first expressed in terms of the change in computed head between successive iterations. The set of equations corresponding to the slice is then solved directly by Gaussian elimination, treating the terms for adjacent slices as known quantities (that is, inserting the most recently computed values of head for the adjacent slices as "known" values in the equations for the slice being processed). The values of head change computed for the slice in this Gaussian elimination process are then each multiplied by an acceleration parameter, ω, generally taken between 1 and 2; the results are taken as the final values of head change in that iteration

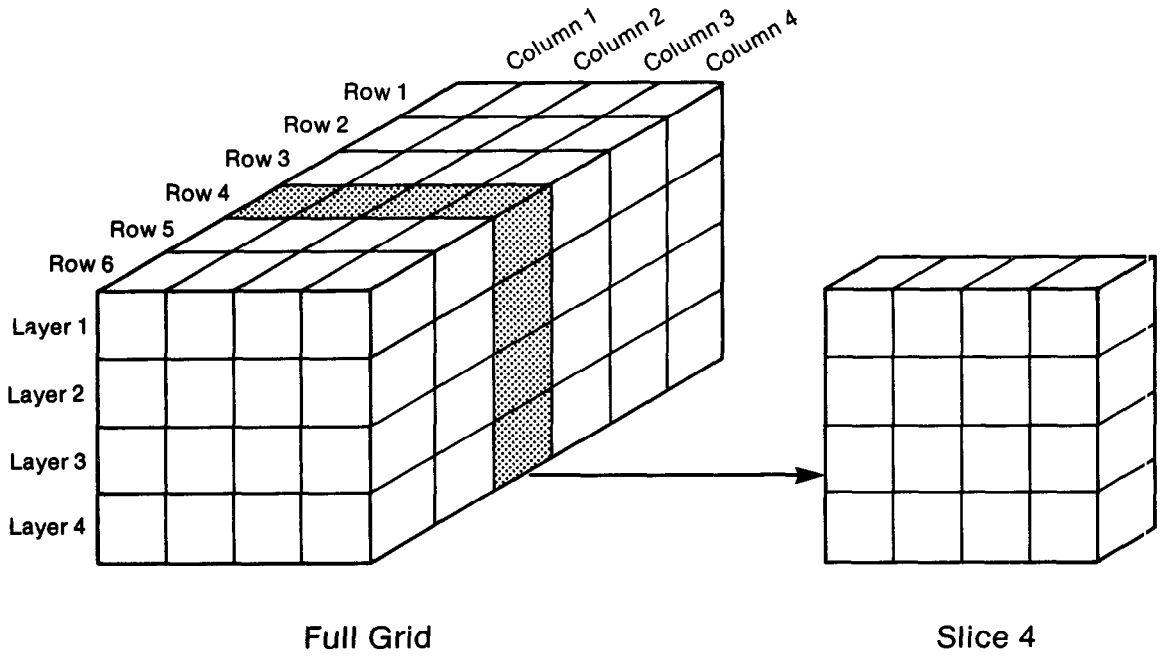


Figure 54.—Division of the three-dimensional model array into vertical slices for processing in the SSOR package.

for the slice. They are added to the respective head values from the preceding iteration to obtain the final estimates of head for the iteration, for that slice. This procedure is repeated for each slice in sequence until all of the slices in the three-dimensional array have been processed, thus completing a single iteration. The entire sequence is then repeated, in successive passes through the series of slices, until the differences between the head values computed in successive iterations is less than the closure criterion at all nodes in the mesh.

It should be noted that even though a direct method of solution (Gaussian elimination) is used within each iteration to process the equations for each individual slice, the overall solution procedure is not direct but iterative. Each direct solution produces only interim values or estimates of head change based on the most recently computed heads in adjacent slices; as successive slices are processed, the computed values continue to change until closure is achieved.

The process of solution described above can be illustrated in more detail through consideration of the node equations. The equation of flow for an individual cell, as developed in chapter 2, is reproduced below with the addition of a second superscript to indicate iteration level

$$
\begin{aligned}
&CV_{i,j,k-1/2}{}^{h}{}_{i,j,k-1}^{m,\ell} + CC_{i-1/2,j,k}{}^{h}{}_{i-1,j,k}^{m,\ell} + CR_{i,j-1/2,k}{}^{h}{}_{i,j-1,k}^{m,\ell} \\
&+ (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
&- CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k}){}^{h}{}_{i,j,k}^{m,\ell} + CR_{i,j+1/2,k}{}^{h}{}_{i,j+1,k}^{m,\ell} \\
&+ CC_{i+1/2,j,k}{}^{h}{}_{i+1,j,k}^{m,\ell} + CV_{i,j,k+1/2}{}^{h}{}_{i,j,k+1}^{m,\ell} = RHS_{i,j,k}
\end{aligned}
\qquad (113)
$$

In equation (113), the superscript m refers to the time step, while the superscript ℓ refers to the iteration level. If an equation of the form of (113) is written for the following iteration level, $\ell+1$, and the left side of equation (113) is then subtracted from each side of the new equation, the result can be written as

$$
\begin{aligned}
&CV_{i,j,k-1/2},(h^{m,\ell+1}_{i,j,k-1}-h^{m,\ell}_{i,j,k-1}) + CC_{i-1/2,j,k}(h^{m,\ell+1}_{i-1,j,k}-h^{m,\ell}_{i-1,j,k}) \\
&+ CR_{i,j-1/2,k}(h^{m,\ell+1}_{i,j-1,k}-h^{m,\ell}_{i,j-1,k}) + (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} \\
&- CR_{i,j-1/2,k} - CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} \\
&+ HCOF_{i,j,k})(h^{m,\ell+1}_{i,j,k}-h^{m,\ell}_{i,j,k}) + CR_{i,j+1/2,k}(h^{m,\ell+1}_{i,j+1,k}-h^{m,\ell}_{i,j+1,k}) \\
&+ CC_{i+1/2,j,k}(h^{m,\ell+1}_{i+1,j,k}-h^{m,\ell}_{i+1,j,k}) + CV_{i,j,k+1/2}(h^{m,\ell+1}_{i,j,k+1}-h^{m,\ell}_{i,j,k+1}) = \\
&RHS_{i,j,k} - CV_{i,j,k-1/2}h^{m,\ell}_{i,j,k-1} - CC_{i-1/2,j,k}h^{m,\ell}_{i-1,j,k} \\
&- CR_{i,j-1/2,k}h^{m,\ell}_{i,j-1,k} - (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} \\
&- CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h^{m,\ell}_{i,j,k} \\
&- CR_{i,j+1/2,k}h^{m,\ell}_{i,j+1,k} - CC_{i+1/2,j,k}h^{m,\ell}_{i+1,j,k} - CV_{i,j,k+1/2}h^{m,\ell}_{i,j,k+1}
\end{aligned}
\tag{114}
$$

In equation (114) the unknown terms are taken as the changes in computed head between iteration ℓ and iteration $\ell+1$--for example, $(h^{m,\ell+1}_{i,j,k} - h^{m,\ell}_{i,j,k})$. Note that when the ℓth iteration has been completed, the right hand side of (114) consists entirely of known terms--it includes the RHS and conductance terms assembled in the formulation process, and estimates of head already obtained during iteration ℓ.

Now suppose that we divide the mesh into vertical slices taken along rows, as shown in figure 54, and isolate the equations associated with the nodes of an individual slice--for example, slice 4 of figure 54, which is taken along row 4 of the three dimensional array. In terms of equation (114), if we are processing slice i, corresponding to row i, we retain the head changes at nodes within this slice as unknown terms, but consider the head changes at nodes in the two adjacent slices to be known values. Thus the terms $CC_{i-1/2,j,k}\ (h^{m,\ell+1}_{i-1,j,k} - h^{m,\ell}_{i-1,j,k})$ and $CC_{i+1/2,j,k}\ (h^{m,\ell+1}_{i+1,j,k} - h^{m,\ell}_{i+1,j,k})$, on the left side of equation (114), are treated as known quantities. If we move these two expressions to the right side of the equation and rearrange, we find that the terms in $h^{m,\ell}_{i-1,j,k}$ and and $h^{m,\ell}_{i+1,j,k}$ drop out, leaving

$$
\begin{aligned}
&CV_{i,j,k-1/2}\,(h^{m,\ell+1}_{i,j,k-1} - h^{m,\ell}_{i,j,k-1}) + CR_{i,j-1/2,k}(h^{m,\ell+1}_{i,j-1,k} - h^{m,\ell}_{i,j-1,k}) \\
&+ (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
&- CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})(h^{m,\ell+1}_{i,j,k} - h^{m,\ell}_{i,j,k}) \\
&+ CR_{i,j+1/2,k}(h^{m,\ell+1}_{i,j+1,k} - h^{m,\ell}_{i,j+1,k}) + CV_{i,j,k+1/2}(h^{m,\ell+1}_{i,j,k+1} - h^{m,\ell}_{i,j,k+1}) = \\
&RHS_{i,j,k} - CV_{i,j,k-1/2}h^{m,\ell}_{i,j,k-1} - CC_{i-1/2,j,k}h^{m,\ell+1}_{i-1,j,k} \\
&- CR_{i,j-1/2,k}h^{m,\ell}_{i,j-1,k} - (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} \\
&- CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h^{m,\ell}_{i,j,k} \\
&- CR_{i,j+1/2,k}h^{m,\ell}_{i,j+1,k} - CC_{i+1/2,j,k}h^{m,\ell+1}_{i+1,j,k} - CV_{i,j,k+1/2}h^{m,\ell}_{i,j,k+1}
\end{aligned}
\qquad (115)
$$

Now suppose the slices are processed in the order of increasing row number, i; then calculations for slice i-1 will be completed in each iteration before calculations for slice i are initiated. It follows that a value of $h^{m,\ell+1}_{i-1,j,k}$ will be available when the processing of slice i is initiated in iteration $\ell+1$, whereas a value of $h^{m,\ell+1}_{i+1,j,k}$ will not be available. Thus the term $CC_{i-1/2,j,k}h^{m,\ell+1}_{i-1,j,k}$ can be incorporated directly as a known term in the processing of slice i, but the term $CC_{i+1/2,j,k}h^{m,\ell+1}_{i+1,j,k}$ cannot. To circumvent this difficulty, the value of $h^{m}_{i+1,j,k}$ from the preceding iteration, $h^{m,\ell}_{i+1,j,k}$, is substituted for $h^{m,\ell+1}_{i+1,j,k}$ on the right side of (115). (Thus in effect we are using the most recently calculated value of head for each adjacent slice.) The resulting equation is

$$
\begin{aligned}
&CV_{i,j,k-1/2}(\tilde{h}^{m,\ell+1}_{i,j,k-1}-h^{m,\ell}_{i,j,k-1}) + CR_{i,j-1/2,k}(\tilde{h}^{m,\ell+1}_{i,j-1,k}-h^{m,\ell}_{i,j-1,k})\\
&+ (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k}\\
&- CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})(\tilde{h}^{m,\ell+1}_{i,j,k}-h^{m,\ell}_{i,j,k})\\
&+ CR_{i,j+1/2,k}(\tilde{h}^{m,\ell+1}_{i,j+1,k}-h^{m,\ell}_{i,j+1,k}) + CV_{i,j,k+1/2}(\tilde{h}^{m,\ell+1}_{i,j,k+1}-h^{m,\ell}_{i,j,k+1}) =\\
&RHS_{i,j,k} - CV_{i,j,k-1/2}h^{m,\ell}_{i,j,k-1} - CC_{i-1/2,j,k}h^{m,\ell+1}_{i-1,j,k}\\
&- CR_{i,j-1/2,k}h^{m,\ell}_{i,j-1,k} - (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k}\\
&- CR_{i,j+1/2,k} - CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h^{m,\ell}_{i,j,k}\\
&- CR_{i,j+1/2,k}h^{m,\ell}_{i,j+1,k} - CC_{i+1/2,j,k}h^{m,\ell}_{i+1,j,k} - CV_{i,j,k+1/2}h^{m,\ell}_{i,j,k+1}
\end{aligned}
\tag{116}
$$

In equation (116), the notation $\tilde{h}$ has been introduced for the head terms in slice i at iteration $\ell+1$. The purpose of this notation will become

clear as the solution process is described. The number of nodes in the slice is NC*NL, where NC is the number of columns in the model and NL the number of layers; and an equation of the form of (116) is formulated at each node. Thus a system of NC*NL equations in NC*NL unknowns is established. Because the number of layers is usually small, the total number of equations is generally small enough so that direct solution by Gaussian elimination is an efficient approach (note that such a procedure would generally not be feasible for the larger set of equations associated with the entire three-dimensional model array.)

The set of equations associated with an individual slice, i, can be written in matrix form as

$$[A]_i\{\Delta\tilde{h}\}_i = \{R\}_i \tag{117}$$

where $[A]_i$ is the coefficient matrix for slice i; $\{\Delta\tilde{h}\}_i$ is a vector of estimates, $\tilde{h}^{m,\ell+1}_{i,j,k} - h^{m,\ell}_{i,j,k}$, for the change in computed head at each node in the slice between iteration ℓ and iteration $\ell+1$; and $\{R\}_i$ is the vector of "constant" terms, representing the right side of equation (116), for slice i.

The Gaussian elimination procedure applied to the matrix equations (117) yields one value of the term $(\tilde{h}^{m,\ell+1}_{i,j,k} - h^{m,\ell}_{i,j,k})$ for each node in the slice. These terms are taken as first estimates for the change in computed head from iteration ℓ to iteration $\ell+1$. Each is multiplied by the acceleration parameter, ω, and each result is added to the corresponding head from the preceding iteration to obtain the final estimate of head for iteration $\ell+1$; that is,

$$h^{m,\ell+1}_{i,j,k} = h^{m,\ell}_{i,j,k} + \omega(\tilde{h}^{m,\ell+1}_{i,j,k} - h^{m,\ell}_{i,j,k}) \tag{118}$$

When values of $h^{m,\ell+1}_{i,j,k}$ have been computed for each node (j,k) in slice i, the procedure of calculation is initiated for the succeeding slice, i+1. When all slices have been processed the iteration is complete, and calculations are initiated for the next iteration unless closure has been achieved.

As illustrated in figure 55-a, the matrix of coefficients $[A]_i$ of equation (117) is symmetric and banded, with a maximum half-bandwidth equal to the number of layers. Because of the symmetry of the matrix, only the lower triangular portion has to be stored; this storage is provided in the program in a two-dimensional array, as illustrated in figure 55-b, with dimensions NL*NC and NL+1. In this example, NL=NC=3.

Adjustment of the acceleration parameter is frequently necessary in SSOR to achieve optimal rates of convergence. For this purpose, methods similar to the trial and error procedure described in Chapter 12, for adjustment of the SIP "seed" value can be applied.

a_{11}	a_{12}		a_{14}					
a_{12}	a_{22}	a_{23}		a_{25}				
	a_{23}	a_{33}			a_{36}			
a_{14}			a_{44}	a_{45}		a_{47}		
	a_{25}		a_{45}	a_{55}	a_{56}			
		a_{36}		a_{56}	a_{66}		a_{68}	
			a_{47}			a_{77}		
					a_{68}		a_{88}	a_{89}
							a_{89}	a_{99}

(a) Coefficient matrix for an individual slice

a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{66}	a_{77}	a_{88}	a_{99}
a_{12}	a_{23}		a_{45}	a_{56}			a_{89}	
					a_{68}			
a_{14}	a_{25}	a_{36}	a_{47}					

(b) Two dimensional array for storage of matrix elements

Figure 55.—Coefficient matrix for slice equations and corresponding computer storage array.

Slice-Successive Overrelaxation Package Input

Input to the Slice-Successive Overrelaxation (SOR) Package is read from the unit specified in IUNIT(11).

FOR EACH SIMULATION

SOR1AL

1. Data: MXITER
   Format: I10

SOR1RP

2. Data: ACCL HCLOSE IPRSOR
   Format: F10.0 F10.0 I10

Explanation of Fields Used in Input Instructions

MXITER--is the maximum number of iterations allowed in a time step.

ACCL--is the acceleration parameter, usually between 1.0 and 2.0.

HCLOSE--is the head change criterion for convergence. When the maximum absolute value of head change from all nodes during an iteration is less than or equal to HCLOSE, iteration stops.

IPRSOR--is the printout interval for SOR. IF IPRSOR is equal to zero, it is changed to 999. The maximum head change (positive or negative) is printed for each iteration of a time step whenever the time step is an even multiple of IPRSOR. This printout also occurs at the end of each stress period regardless of the value of IPRSOR.

Module Documentation for the Slice-Successive Overrelaxation Package

The Slice-Successive Overrelaxation Package (SOR1) consists of three primary modules and one submodule. They are:

Primary Modules

SOR1AL	Allocates space for arrays.
SOR1RP	Reads control information needed by the SOR1 Package.
SOR1AP	Performs one iteration of slice-successive overrelaxation.

Submodule

SSOR1B	Solves a system of linear equations.

Narrative for Module SOR1AL

Module SOR1AL allocates space in the X array for SOR arrays. The SOR arrays are A, RES, IEQPNT, HDCG, and LRCH. "A" holds the main diagonal and the lower diagonals of the symmetric coefficient matrix for a single slice. RES holds the residual vector (the right hand sides) for a single slice. IEQPNT holds a sequential identification number for each cell in a slice. HDCG holds the maximum head change for each iteration. LRCH holds the location of the cell (row, column, and layer) which had the maximum head change for each iteration.

Module SOR1AL performs its functions in the following order:

1. Print a message identifying the SOR Package.

2. Read and print the maximum number of iterations.

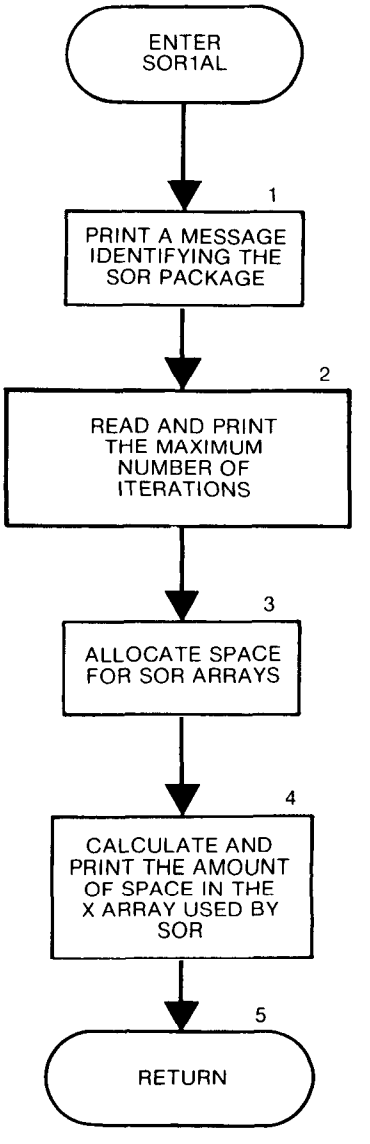
3. Allocate the required space in the X array. The X-array location pointer (ISUM) is saved in the variable ISOLD prior to allocation so that the space required for SOR can be calculated in step 4. To allocate space for an array, the array-location variable is set equal to ISUM. Then ISUM is incremented by the required number of elements.

4. Calculate and print the space used in the X array. The space used by SOR is ISUM - ISOLD.

5. RETURN

Flow Chart for Module SOR1AL

X array is the pool of memory space from which space is allocated for arrays used by various packages.



```
      SUBROUTINE SOR1AL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,LCIEQP,
     1        MXITER,NCOL,NLAY,NSLICE,MBW,IN,IOUT)
C
C-----VERSION 1638 24JUL1987 SOR1AL
C     ******************************************************************
C     ALLOCATE STORAGE FOR SOR ARRAYS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------PRINT A MESSAGE IDENTIFYING SOR PACKAGE
      WRITE(IOUT,1)IN
    1 FORMAT(1HO,'SOR1 -- SLICE-SUCCESSIVE OVERRELAXATION PACKAGE'
     1,', VERSION 1, 9/1/87 INPUT READ FROM UNIT',I3)
C
C2------READ AND PRINT MXITER (MAXIMUM # OF ITERATIONS)
      READ(IN,2) MXITER
    2 FORMAT(I10)
      WRITE(IOUT,3) MXITER
    3 FORMAT(1X,I5,' ITERATIONS ALLOWED FOR SOR CLOSURE')
C
C3------ALLOCATE SPACE FOR THE SOR ARRAYS
      ISOLD=ISUM
      NSLICE=NCOL*NLAY
      MBW=NLAY+1
      LCA=ISUM
      ISUM=ISUM+NSLICE*MBW
      LCRES=ISUM
      ISUM=ISUM+NSLICE
      LCIEQP=ISUM
      ISUM=ISUM+NSLICE
      LCHDCG=ISUM
      ISUM=ISUM+MXITER
      LCLRCH=ISUM
      ISUM=ISUM+3*MXITER
      ISP=ISUM-ISOLD
C
C4------CALCULATE AND PRINT THE SPACE USED IN THE X ARRAY
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED BY SOR')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module SOR1AL

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.
ISOLD	Package	Before this module allocates space, ISOLD is set equal to ISUM. After allocation, ISOLD is subtracted from ISUM to get ISP, the amount of space in the X array allocated by this module.
ISP	Module	Number of words in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	Index number of the last element of the X array allocated by this module.
LCA	Package	Location in the X array of the first element of array A.
LCHDCG	Package	Location in the X array of the first element of array HDCG.
LCIEQP	Package	Location in the X array of the first element of array IEQPNT.
LCLRCH	Package	Location in the X array of the first element of array LRCH.
LCRES	Package	Location in the X array of the first element of array RES.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
MBW	Package	Maximum bandwidth of the coefficient matrix +1.
MXITER	Package	Maximum number of iterations.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NSLICE	Package	Number of cells in a slice.

Narrative for Module SOR1RP

Module SOR1RP reads data for the SOR package: the acceleration parameter (ACCL), also called the relaxation factor; the closure criterion (HCLOSE); and the time-step interval (IPRSOR) for printing head change. This module does not have a flow chart. Module SOR1RP performs its functions in the following order:

1. Read the acceleration parameter (ACCL), the closure criterion (HCLOSE), and the interval for printing head change (IPRSOR). If ACCL is zero, substitute a default value of 1.0. If IPRSOR is less than one, set it equal to 999.

2. Print the maximum number of iterations (MXITER), the acceleration parameter (ACCL), the closure criterion (HCLOSE), and the head-change interval (IPRSOR).

3. RETURN.

```
      SUBROUTINE SOR1RP(MXITER,ACCL,HCLOSE,IN,IPRSOR,IOUT)
C
C
C-----VERSION 1005 16MAR1983 SOR1RP
C     ******************************************************************
C     READ PARAMETERS FOR SOR
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------READ THE ACCELERATION PARAMETER/RELAXATION FACTOR (ACCL) THE
C1------CLOSURE CRITERION (HCLOSE) AND THE NUMBER OF TIME STEPS
C1------BETWEEN PRINTOUTS OF MAXIMUM HEAD CHANGES (IPRSOR).
      READ(IN,1) ACCL,HCLOSE,IPRSOR
    1 FORMAT(2F10.0,I10)
      IF(ACCL.EQ.0.) ACCL=1.
      IF(IPRSOR.LT.1) IPRSOR=999
C
C2------PRINT ACCL, HCLOSE, IPRSOR
      WRITE(IOUT,100)
  100 FORMAT(1H0,///57X,'SOLUTION BY SLICE-SUCCESSIVE OVERRELAXATION'
     1/57X,43('-'))
      WRITE(IOUT,115) MXITER
  115 FORMAT(1H0,47X,'MAXIMUM ITERATIONS ALLOWED FOR CLOSURE =',I9)
      WRITE(IOUT,120) ACCL
  120 FORMAT(1H ,63X,'ACCELERATION PARAMETER =',G15.5)
      WRITE(IOUT,125) HCLOSE
  125 FORMAT(1H ,52X,'HEAD CHANGE CRITERION FOR CLOSURE =',E15.5)
      WRITE(IOUT,130) IPRSOR
  130 FORMAT(1H ,52X,'SOR HEAD CHANGE PRINTOUT INTERVAL =',I9)
C
C3------RETURN
      RETURN
      END
```

List of Variables for Module SOR1RP

Variable	Range	Definition
ACCL	Package	Acceleration parameter.
HCLOSE	Package	Closure criterion for the iterative procedure.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRSOR	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
MXITER	Package	Maximum number of iterations.

Narrative for Module SOR1AP

Module SOR1AP performs one iteration of the Slice-Successive Over-relaxation (SSOR) algorithm for solving the system of finite-difference equations. The conductances CC, CR, and CV and the composite terms HCOF and RHS (see equation (27)) which are calculated by the formulation procedure are combined, row by row (slice by slice), to form the coefficient matrix [A] and the vector {RES} on the right hand side of the matrix equation for a single slice. Since the coefficient matrix is symmetric and banded, only main diagonals and NLAY subdiagonals are saved. As heads are calculated, they are stored in the array HNEW. The matrix [A] and the vector {RES} are passed to a submodule SSOR1B which solves the matrix equation for a vector of approximate head changes which is then multiplied by the relaxation factor to get the final head changes for the iteration. The final head changes are added to the heads from the preceding iteration to get the heads for the current iteration. The final head changes for the iteration are compared to the closure criterion to see if the iterative procedure has closed.

Module SOR1AP performs its functions in the following order:

1. Calculate the number of elements in the compressed coefficient matrix [A].

2. Process the slices (rows) one at a time (DO STEPS 3-7).

3. Clear the A array.

4. Assign integers sequentially to the active cells in the slice (remember that finite-difference equations are formulated only for active cells).

5. Calculate the elements in the compressed coefficient matrix [A] and the residual vector {RES}. Process the cells in the slice one cell at a time.

If the cell is inactive, move on to the next cell. The elements in the main diagonal of the coefficient matrix (the multipliers of $h_{i,j,k}$) will consist of HCOF plus conductances to the six adjacent cells. They will be formed in an accumulator called EE. The contents of EE multiplied by the head from the previous iteration (HNEW) are subtracted from an accumulator (R) to form the residual.

(a) Determine the equation number (NEQ) of the cell. If NEQ is zero, the cell is inactive. Move on to the next cell.

(b) Set the accumulators EE and R equal to HCOF and RHS, respectively. Note: HNEW contains head from the last iteration.

(c) If there is a node to the left, subtract the conductance from EE and subtract the conductance times HNEW from R.

(d) If there is a node to the right, subtract the conductance from EE, and subtract the conductance times HNEW from R; and, if the cell to the right is active, move the conductance into the compressed coefficient matrix [A]. Remember that the coefficient matrix is symmetric so the conductance to the left in step 5(c) did not have to be stored.

(e) If there is a node to the rear, subtract the conductance from EE and subtract the conductance times HNEW from R.

(f) If there is a node to the front, subtract the conductance from EE and subtract the conductance times HNEW from R. Remember that the

form of the SSOR equations does not have terms containing head in adjacent rows on the left hand side.

(g) If there is a node above, subtract the conductance from EE and subtract the conductance times HNEW from R.

(h) If there is a node below, subtract the conductance from EE and subtract the conductance times HNEW from R; and, if the cell below is active, move the conductance into A.

(i) Move EE into the first row of A. The first row in A corresponds to the main diagonal in the "full" coefficient matrix. Subtract EE times HNEW from R and store it in the residual vector.

6. If there are no equations for this slice, go on to the next slice. If there is only one equation, solve it directly and leave the result in the residual vector {RES}. If there are two or more equations, call submodule SSOR1B to solve the system of equations for the slice leaving the results (first estimate of head change for this iteration) in the vector {RES}.

7. For each cell in the slice, calculate the head for the current iteration.

(a) Multiply the first estimate of head change for this iteration by the relaxation factor to get the final estimate of head change for this iteration.

(b) Add the final head change for this iteration to the head from the last iteration to get the head for this iteration.

(c) If the head change for this cell is greater than that for any other cell, store the head change and the location of the cell.

8. Save the largest head change from this iteration so that it can be printed at the end of the time step.

9. Compare the biggest head change (BIGG) to the closure criterion (HCLOSE). If HCLOSE is greater than BIGG, set the convergence flag (ICNVG) equal to one.

10. If you have not converged and you have not exceeded the maximum number of iterations, RETURN.

11. Print the number of iterations.

12. If convergence failed, or this is the last time step, or this is the time step interval specified by the user, print the maximum head change for each iteration in this time step.

13. RETURN.

Flow Chart for Module SOR1AP

A is a compressed coefficient matrix for a slice. It contains the main diagonal of the full matrix and the NLAY diagonals below it. (NLAY is the number of layers.)

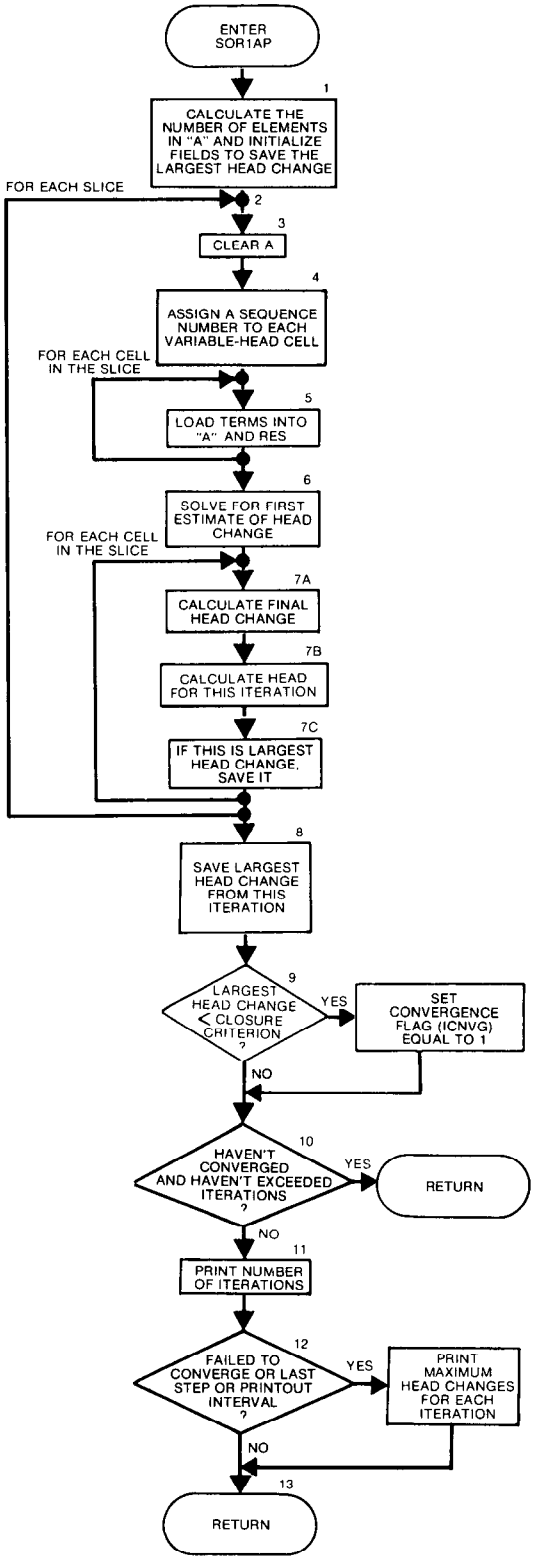
Sequence Number is a number used to identify the internal (variable-head) cells in a slice and also the equations for each internal cell.

RES is a vector containing the residuals for a slice. It consists of RHS (from the basic finite-difference equation) plus all of those terms which are moved to the right hand side to get the equations ready for solution in residual form.

First estimates of head change: these are the head changes calculated by simultaneously solving the equations for a slice. They will be multiplied by the relaxation factor to get final estimates of head change.

Final estimates of head change: these are the head changes calculated by multiplying first estimates by the relaxation factor. They are added to the heads from the previous iteration to get head for the current iteration.

ICNVG is the convergence flag. It is set in the approximator and returned to the MAIN Program so that the iteration loop can be terminated.



```
      SUBROUTINE SOR1AP(HNEW,IBOUND,CR,CC,CV,HCOF,RHS,A,RES,IEQPNT,
     1       HDCG,LRCH,KITER,HCLOSE,ACCL,ICNVG,KSTP,KPER,
     2       IPRSOR,MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C-----VERSION 0936 09MAY1983 SOR1AP
C     ******************************************************************
C     SOLUTION BY SLICE-SUCCESSIVE OVERRELAXATION -- 1 ITERATION
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,DIFF,DP,EE,R,HCFHNW,HHCOF
C
      DIMENSION HNEW(NCOL,NROW,NLAY), IBOUND(NCOL,NROW,NLAY),
     1   CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY),
     1   CV(NCOL,NROW,NLAY), HCOF(NCOL,NROW,NLAY), RHS(NCOL,NROW,NLAY),
     2   HDCG(MXITER), LRCH(3,MXITER),A(MBW,NSLICE),RES(NSLICE),
     3   IEQPNT(NLAY,NCOL)
C     ------------------------------------------------------------------
C
C1------CALCULATE # OF ELEMENTS IN COMPRESSED MATRIX A AND
C1------INITIALIZE FIELDS TO SAVE LARGEST HEAD CHANGE.
      NA=MBW*NSLICE
      BIG=0.
      ABSBIG=0.
      IB=0
      JB=0
      KB=0
C
C2------PROCESS EACH SLICE
      DO 500 I=1,NROW
C
C3------CLEAR A
      DO 110 J=1,NSLICE
      DO 110 K=1,MBW
  110 A(K,J)=0.
C
C4------ASSIGN A SEQUENCE # TO EACH VARIABLE HEAD CELL.
      NEQT=0
      DO 200 J=1,NCOL
      DO 200 K=1,NLAY
      IEQPNT(K,J)=0
      IF(IBOUND(J,I,K).LE.0) GO TO 200
      NEQT=NEQT+1
      IEQPNT(K,J)=NEQT
  200 CONTINUE
C
C5------FOR EACH CELL LOAD MATRIX A AND VECTOR RES
      DO 300 J=1,NCOL
      DO 300 K=1,NLAY
C
C5A-----IF SEQUENCE # IS 0 (CELL IS EXTERNAL) GO ON TO NEXT CELL
      NEQ=IEQPNT(K,J)
      IF(NEQ.EQ.0) GO TO 300
C
C5B-----INITIALIZE ACCUMULATORS EE AND R
      EE=0.
      R=RHS(J,I,K)
C
```

```
C5C-----IF NODE TO LEFT SUBTRACT TERMS FROM EE AND R
      IF(J.EQ.1) GO TO 120
      DP=CR(J-1,I,K)
      R=R-DP*HNEW(J-1,I,K)
      EE=EE-DP
C
C5D-----IF NODE TO RIGHT SUBTRACT TERMS FROM EE & R, MOVE COND TO A
  120 IF(J.EQ.NCOL) GO TO 125
      SP=CR(J,I,K)
      DP=SP
      R=R-DP*HNEW(J+1,I,K)
      EE=EE-DP
      NXT=IEQPNT(K,J+1)
      IF(NXT.GT.0) A(1+NXT-NEQ,NEQ)=SP
C
C5E-----IF NODE TO REAR SUBTRACT TERMS FROM EE AND R
  125 IF(I.EQ.1) GO TO 130
      DP=CC(J,I-1,K)
      R=R-DP*HNEW(J,I-1,K)
      EE=EE-DP
C
C5F-----IF NODE TO FRONT SUBTRACT TERMS FROM EE AND R
  130 IF(I.EQ.NROW) GO TO 132
      DP=CC(J,I,K)
      R=R-DP*HNEW(J,I+1,K)
      EE=EE-DP
C
C5G-----IF NODE ABOVE SUBTRACT TERMS FROM EE AND R
  132 IF(K.EQ.1) GO TO 134
      DP=CV(J,I,K-1)
      R=R-DP*HNEW(J,I,K-1)
      EE=EE-DP
C
C5H-----IF NODE BELOW SUBTRACT TERMS FROM EE & R AND MOVE COND TO A
  134 IF(K.EQ.NLAY) GO TO 136
      SP=CV(J,I,K)
      DP=SP
      R=R-DP*HNEW(J,I,K+1)
      EE=EE-DP
      IF(IEQPNT(K+1,J).GT.0) A(2,NEQ)=SP
C
C5I-----MOVE EE INTO A, SUBTRACT EE TIMES LAST HEAD FROM R TO GET RES
  136 HHCOF=HCOF(J,I,K)
      A(1,NEQ)=EE+HHCOF
      HNW=HNEW(J,I,K)
      HCFHNW=HNW*HCOF(J,I,K)
      RES(NEQ)=R-EE*HNEW(J,I,K)-HCFHNW
  300 CONTINUE
C
C6------IF NO EQUATIONS GO TO NEXT SLICE, IF ONE EQUATION SOLVE
C6------DIRECTLY, IF 2 EQUATIONS CALL SSOR1B TO SOLVE FOR FIRST
C6------ESTIMATE OF HEAD CHANGE FOR THIS ITERATION.
      IF(NEQT.LT.1) GO TO 500
      IF(NEQT.EQ.1) RES(1)=RES(1)/A(1,1)
      IF(NEQT.GE.2) CALL SSOR1B(A,RES,NEQT,NA,MBW)
C
C7------FOR EACH CELL IN SLICE CALCULATE FINAL HEAD CHANGE THEN HEAD.
      DO 400 J=1,NCOL
      DO 400 K=1,NLAY
      NEQ=IEQPNT(K,J)
      IF(NEQ.EQ.0) GO TO 400
C
```

```
C7A-----MULTIPLY FIRST ESTIMATE OF HEAD CHANGE BY RELAX FACTOR TO
C7A-----GET FINAL ESTIMATE OF HEAD CHANGE FOR THIS ITERATION.
      DH=RES(NEQ)*ACCL
      DIFF=DH
C
C7B-----ADD FINAL ESTIMATE TO HEAD FROM LAST ITERATION TO GET HEAD
C7B-----FOR THIS ITERATION.
      HNEW(J,I,K)=HNEW(J,I,K)+DIFF
C
C7C-----SAVE FINAL HEAD CHANGE IF IT IS THE LARGEST
      ABSDH=ABS(DH)
      IF(ABSDH.LE.ABSBIG) GO TO 400
      ABSBIG=ABSDH
      BIG=DH
      IB=I
      JB=J
      KB=K
  400 CONTINUE
C
C
  500 CONTINUE
C
C8------SAVE LARGEST HEAD CHANGE FOR THIS ITERATION
      HDCG(KITER)=BIG
      LRCH(1,KITER)=KB
      LRCH(2,KITER)=IB
      LRCH(3,KITER)=JB
C
C9------IF LARGEST HEAD CHANGE IS SMALLER THAN CLOSURE THEN SET
C9------CONVERGE FLAG (ICNVG) EQUAL TO 1.
      ICNVG=0
      IF(ABSBIG.LE.HCLOSE) ICNVG=1
C
C10-----IF NOT CONVERGED AND NOT EXCEDED ITERATIONS THEN RETURN
      IF(ICNVG.EQ.0 .AND. KITER.NE.MXITER) RETURN
      IF(KSTP.EQ.1) WRITE(IOUT,600)
  600 FORMAT(1H0)
C
C11-----PRINT NUMBER OF ITERATIONS
      WRITE(IOUT,601) KITER,KSTP,KPER
  601 FORMAT(1X,I5,' ITERATIONS FOR TIME STEP',I4,' IN STRESS PERIOD',
     1        I3)
C
C12-----IF FAILED TO CONVERGE OR LAST TIME STEP OR PRINTOUT
C12-----INTERVAL SPECIFIED BY USER IS HERE THEN PRINT MAXIMUM
C12-----HEAD CHANGES FOR EACH ITERATION.
      IF(ICNVG.NE.0 .AND. KSTP.NE.NSTP .AND. MOD(KSTP,IPRSOR).NE.0)
     1      GO TO 700
      WRITE(IOUT,5)
    5 FORMAT(1H0,'MAXIMUM HEAD CHANGE FOR EACH ITERATION:'/
     1    1H0,4('    HEAD CHANGE  LAYER,ROW,COL')/1X,120('-'))
      WRITE (IOUT,10) (HDCG(J),(LRCH(I,J),I=1,3),J=1,KITER)
   10 FORMAT((1X,4(4X,G12.4,' (',I3,',',I3,',',I3,')')))
      WRITE(IOUT,11)
   11 FORMAT(1H0)
C
C13-----RETURN
  700 RETURN
C
      END
```

List of Variables for Module SOR1AP

Variable	Range	Definition
A	Package	DIMENSION (MBW,NSLICE), Compressed coefficient matrix for a slice.
ABSDIG	Module	Largest ABSDH for this iteration.
ABSDH	Module	Absolute value of head change in a cell for the current iteration.
ACCL	Package	Acceleration parameter.
CC	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the column direction. CC(J,I,K) contains conductance between nodes (J,I,K) and (J,I+1,K).
CR	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the row direction. CR(J,I,K) contains conductance between nodes (J,I,K) and (J+1,I,K).
CV	Global	DIMENSION (NCOL,NROW,NLAY-1), Conductance in the vertical direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DH	Module	Change in head in a cell during one iteration.
DIFF	Module	Double-precision change in head (DH).
DP	Module	Double-precision temporary field.
EE	Module	Main diagonal term in the finite-difference equation.
HCLOSE	Package	Closure criterion for the iterative procedure.
HCOF	Global	DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the cell (J,I,K) in the finite-difference equation.
HDCG	Package	DIMENSION (MXITER), Maximum head change for each iteration.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell. HNEW changes at each iteration.
I	Module	Index for rows.
IB	Module	Row number of the cell containing the largest head change.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. < 0, constant-head cell = 0, inactive cell > 0, variable-head cell
ICNVG	Global	Flag is set equal to one when the iteration procedure has converged.
IEQPNT	Global	DIMENSION (NLAY,NCOL), Sequence numbers for variable-head cells in a slice.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRSOR	Package	Frequency (in time steps) with which the maximum head changes for each iteration will be printed.
J	Module	Index for columns.
JB	Module	Column number of the cell containing the largest head change.
K	Module	Index for layers.
KB	Module	Layer number of the cell containing the largest head change.
KITER	Global	Iteration counter. Reset at the start of each time step.
KPER	Global	Stress period counter.

List of Variables for Module SOR1AP (Continued)

Variable	Range	Definition
KSTP	Global	Time step counter. Reset at the start of each stress period.
LRCH	Package	DIMENSION (MXITER), Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MBW	Package	Maximum bandwidth of the coefficient matrix +1.
MXITER	Package	Maximum number of iterations.
NA	Package	Number of elements in the compressed coefficient matrix (A).
NCOL	Global	Number of columns in the grid.
NEQ	Module	Index for equations (variable-head cells) in a slice.
NEQT	Package	Number of equations (variable-head cells) in a slice.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NSLICE	Package	Number of cells in a slice.
NSTP	Global	Number of time steps in the current stress period.
NXT	Module	Sequence number of the cell to the right.
R	Module	Right hand side of the finite-difference equation as modified (terms for the adjacent rows moved to the right) for solution by the slice-successive overrelaxation.
RES	Package	DIMENSION (NSLICE), Residual.
RHS	Global	DIMENSION (NCOL,NROW,NLAY), Right hand side of the finite-difference equation. RHS is an accumulation of terms from several different packages.
SP	Module	Single-precision temporary field.

Narrative for Module SSOR1B

Module SSOR1B uses Gaussian elimination to solve a matrix equation of the form [A]{X}={B}. The matrix [A] is symmetric and banded with rank "N." It is stored in the compressed format shown in figure 55 and uses a single subscript. The vector {X}, as it is calculated, is stored in the space reserved for vector {B}.

The indices used in the module flow chart are those for standard matrix organization. The indices actually used in the program are based on the compressed format and a single index. Module SSOR1B performs its functions in the following order:

1. Work through the first N-1 rows using each one, in sequence, as the pivot row (row I).

2. Calculate the inverse of the main diagonal element---$a_{I,I}$. (The index (ID) points to $a_{I,I}$ which is the first element of column I in the compressed matrix [A].)

3. Modify each of the rows after row I so that the terms corresponding to the pivot term are eliminated. Since the coefficient matrix is banded, there are only MBW-1 equations (where MBW is the maximum half-bandwidth plus one) where the term to be eliminated is not already equal to zero. The rows are indexed by "L." (The equation corresponding to row "L" is referred to as equation "L.")

4. Calculate the coefficient C which when multiplied by the pivot equation and subtracted from equation L will eliminate a term in equation L.

(The index IB points to a coefficient in the pivot equation which, because of symmetry, is equal to the coefficient to be eliminated.)

5. Calculate the new coefficients in equation L for each of the terms to the right of the coefficient that is being eliminated. Because the matrix is banded, there are only MBW-1 nonzero terms to the right of the pivot. Therefore, at most, MBW-1 coefficients have to be calculated.

6. Subtract C times a coefficient in the pivot equation from the corresponding coefficient in equation L.

7. Subtract C times the right side of the pivot equation from the right side of equation L. (The index LB points to the coefficient in equation L which must be calculated.)

8. Solve equation N for X(N) putting the result in B(N).

9. Work backward from equation N-1 solving each equation (equation L) for X(L).

10. Set the accumulator "SUM" equal to zero.

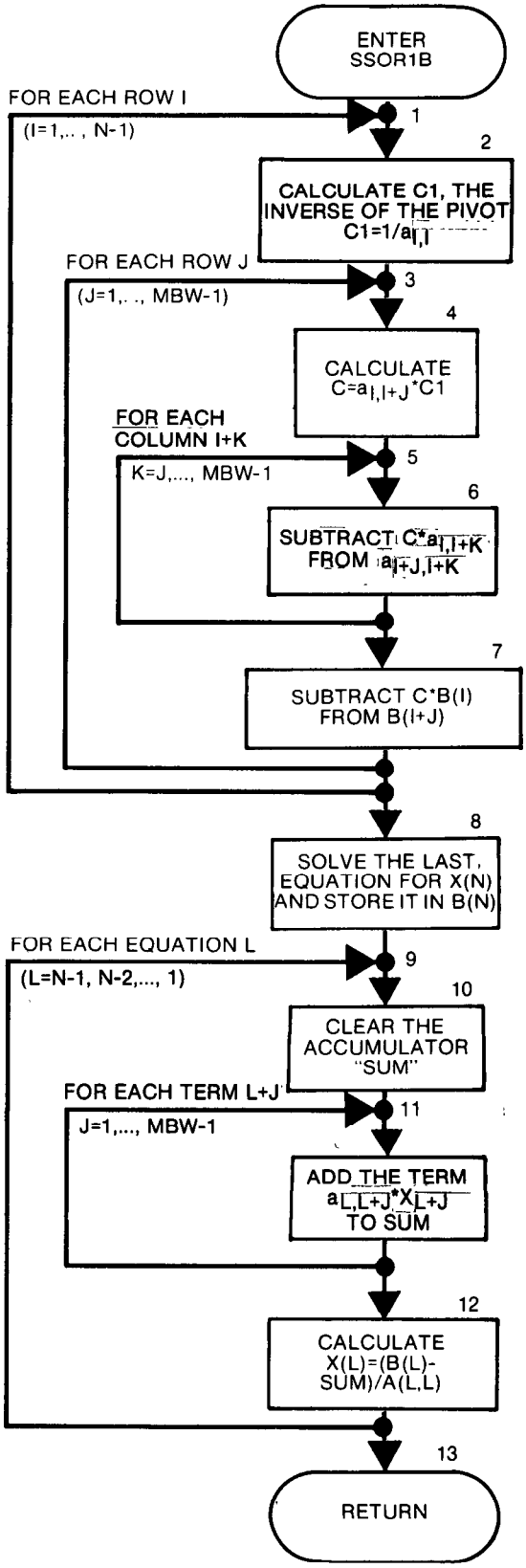
11. Multiply the coefficient to the right of the main diagonal (in equation L) by the corresponding value of X and add it to the sum.

12. Calculate the value of X(L) and store it in B(L).

13. RETURN.

Flow Chart for Module SSOR1B

MBW is the maximum half bandwidth
of the coefficient matrix plus one.



```
      SUBROUTINE SSOR1B(A,B,N,NA,MBW)
C
C
C-----VERSION 1359 31MAR1983 SSOR1B
C     ******************************************************************
C     SOLVE A SYMMETRIC SET OF EQUATIONS
C        A IS COEFFICIENT MATRIX IN COMPRESSED FORM
C        B IS RIGHT HAND SIDE AND IS REPLACED BY SOLUTION
C        N IS NUMBER OF EQUATIONS TO BE SOLVED
C        MBW IS BANDWIDTH OF A
C        NA IS ONE-DIMENSION SIZE OF A
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION A(NA),B(N)
C     ------------------------------------------------------------------
C
      NM1=N-1
      MBW1=MBW-1
      ID=1-MBW
C
C1------SEQUENTIALLY USE EACH OF THE FIRST N-1 ROWS AS
C1------THE PIVOT ROW.
      DO 20 I=1,NM1
C
C2------CALCULATE THE INVERSE OF THE PIVOT.
      ID=ID+MBW
      C1=1./A(ID)
      LD=ID
      L=I
C
C3------FOR EACH ROW AFTER THE PIVOT ROW (THE TARGET ROW)
C3------ELIMINATE THE COLUMN CORRESPONDING TO THE PIVOT.
      DO 15 J=1,MBW1
      L=L+1
      IF(L.GT.N) GO TO 20
      IB=ID+J
C
C4------CALCULATE THE FACTOR NEEDED TO ELIMINATE A TERM IN THE
C4------TARGET ROW.
      C=A(IB)*C1
      LD=LD+MBW
```

```
      LB=LD-1
C
C5------MODIFY THE REST OF THE TERMS IN THE TARGET ROW.
      DO 10 K=J,MBW1
C
C6------SUBTRACT THE FACTOR TIMES A TERM IN THE PIVOT ROW
C6------FROM THE CORRESPONDING COLUMN IN THE TARGET ROW.
      LB=LB+1
      A(LB)=A(LB)-C*A(ID+K)
   10 CONTINUE
C
C7------MODIFY THE RIGHT SIDE OF THE EQUATION CORRESPONDING
C7------TO THE TARGET ROW.
      B(I+J)=B(I+J)-C*B(I)
   15 CONTINUE
   20 CONTINUE
      ID=ID+MBW
C
C8------SOLVE THE LAST EQUATION.
      B(N)=B(N)/A(ID)
C
C9------WORKING BACKWARDS SOLVE THE REST OF THE EQUATIONS.
      DO 70 I=1,NM1
      ID=ID-MBW
C
C10-----CLEAR THE ACCUMULATOR SUM.
      SUM=0.0
      L=N-I
      MBW1M=MIN0(MBW1,I)
C
C11-----ADD THE KNOWN TERMS IN EQUATION L TO SUM.
      DO 60 J=1,MBW1M
      SUM=SUM+A(ID+J)*B(L+J)
   60 CONTINUE
C
C12-----SOLVE FOR THE ONE UNKNOWN IN EQUATION L.
      B(L)=(B(L)-SUM)/A(ID)
   70 CONTINUE
C
C13-----RETURN
      RETURN
      END
```

List of Variables for Module SSOR1B

Variable	Range	Definition
A	Package	DIMENSION (MBW,NSLICE), Compressed coefficient matrix for a slice.
B	Package	DIMENSION (N), Right-hand-side vector.
C	Module	Factor needed to eliminate a term in the target row.
C1	Module	Inverse of pivot.
I	Module	Index for rows in the SSOR matrix (not the grid).
IB	Module	Index for elements to the right of the pivot.
ID	Module	Index of pivots.
J	Module	Index for columns.
K	Module	Index for columns.
L	Module	Index for equations.
LB	Module	Index for elements in the target row to the right of the main diagonal.
LD	Module	Index of the main diagonal elements in the target rows.
MBW	Package	Maximum bandwidth of the coefficient matrix +1.
MBW1	Module	Maximum bandwidth of the coefficient matrix.
MBW1M	Module	Maximum possible number of the nonzero elements to the right of the main diagonal.
N	Package	Number of equations to be solved.
NA	Package	One-dimension size of compressed matrix "A."
NM1	Module	N-1.
SUM	Module	In back substitution--in equation L, sum of terms to the right of the main diagonal term (L,L).

CHAPTER 14

UTILITY MODULES

Utility modules are those submodules which perform general tasks common to several different packages. The name of a utility module always consists of a "U" followed by a five-character mnemonic. There are eight utility modules:

UBUDSV	Writes an unformatted record consisting of an array with one real number for each cell in the grid.
ULASAV	Writes an unformatted record consisting of an array with one real number for each cell in a layer.
ULAPRS and ULAPRW	Prints one two-dimensional array which contains one real number for each cell in a layer. ULAPRS prints, in strip form, the first N columns (where N is the number of values that can fit on one print line) of each row and then the next N columns, etc., until all columns of each row are printed (fig. 56). ULAPRW prints, in wrap form, all of row 1, all of row 2, and all of row 3, etc. The format for printing arrays is shown in table 2.
UCOLNO	Prints column numbers at the top of each page of data printed by ULAPRS and ULAPRW.
U2DREL	Reads a two-dimensional array of real numbers.
U2DINT	Reads a two-dimensional array of integers.
U1DREL	Reads a one-dimensional array of real numbers.

	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17			
1	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
2	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
3	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
4	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
5	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
6	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			
7	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79			

WRAP FORM

	1	2	3	4	5	6	7	8	9	10
1	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
2	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
3	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
4	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
5	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
6	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
7	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79

	11	12	13	14	15	16	17
1	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
2	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
3	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
4	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
5	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
6	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79
7	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79	1325.79

STRIP FORM

Figure 56.—Illustration of wrap and strip forms of printed output for a layer containing 7 rows and 17 columns.

Table 2.--Print-format codes for utility modules ULAPRS and ULAPRW.

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

Input Instructions For Array Reading Utility Modules

The real two-dimensional array reader (U2DREL), the integer two-dimensional array reader (U2DINT), and the real one-dimensional array reader (U1DREL) read one array-control record and, optionally, a data array in a format specified on the array-control record. The control record is read from the input unit number specified for the major option that is requesting the array. For example, the Recharge Package uses U2DREL to read the RECH array. The input unit for the recharge option is contained in IUNIT (8), and accordingly, the RECH array control record is read on this input unit.

FOR REAL ARRAY READER (U2DREL or U1DREL)

Data:	LOCAT	CNSTNT	FMTIN	IPRN
Format:	I10	F10.0	5A4	I10

FOR INTEGER ARRAY READER (U2DINT)

Data:	LOCAT	ICONST	FMTIN	IPRN
Format:	I10	I10	5A4	I10

Explanation of Fields Used in Input Instructions

LOCAT--indicates the location of the data which will be put in the array.

If LOCAT < 0, the sign is reversed to give the unit number from which an unformatted record will be read.

If LOCAT = 0, every element in the array will be set equal to the value CNSTNT/ICONST.

If LOCAT > 0, it is the unit number from which data values will be read in the format specified in the third field of the array-control record (FMTIN).

CNSTNT/ICONST--is a constant. Its use depends on the value of LOCAT.

If LOCAT = 0, every element in the array is set equal to CNSTNT/ICONST.

If LOCAT ≠ 0, and if CNSTNT/ICONST ≠ 0, every element in the array is multiplied by CNSTNT/ICONST.

FMTIN--is the format of records containing the array values. It is used only if the first field in the array-control record (LOCAT) contains a positive number. The format must be enclosed in parentheses; for example, (15F5.0) for real data and (15I5) for integer data.

IPRN--is a flag indicating that the array being read should be printed and a code for indicating the format that should be used. It is used only if LOCAT is not equal to zero. The format codes are different for each of the three modules. IPRN is set to zero when the specified value exceeds those defined in the chart below. If IPRN is less than zero, the array will not be printed.

IPRN	U2DREL	U2DINT	U1DREL
0	10G11.4	10I11	10G12.5
1	11G10.3	60I1	
2	9G13.6	40I2	
3	15F7.1	30I3	
4	15F7.2	25I4	
5	15F7.3	20I5	
6	15F7.4		
7	20F5.0		
8	20F5.1		
9	20F5.2		
10	20F5.3		
11	20F5.4		
12	10G11.4		

Narrative for Module UBUDSV

Utility module UBUDSV writes an unformatted record consisting of an array dimensioned (NCOL, NROW, NLAY). The record containing the array is preceded by an unformatted record containing identifying information. The identifying information consists of:

KSTP	current time step	integer	1 word
KPER	current stress period	integer	1 word
TEXT	label	character string	4 words
NCOL	number of columns	integer	1 word
NROW	number of rows	integer	1 word
NLAY	number of layers	integer	1 word

Documentation of this module consists only of comments in the program and a list of variables.

```
      SUBROUTINE UBUDSV(KSTP,KPER,TEXT,IBDCHN,BUFF,NCOL,NROW,NLAY,IOUT)
C
C
C-----VERSION 1637 12MAY1987 UBUDSV
C     ***************************************************************
C     RECORD CELL-BY-CELL FLOW TERMS FOR ONE COMPONENT OF FLOW.
C     ***************************************************************
C
C        SPECIFICATIONS:
C     -------------------------------------------------------------
      CHARACTER*4 TEXT
      DIMENSION TEXT(4),BUFF(NCOL,NROW,NLAY)
C     -------------------------------------------------------------
C
C1------WRITE AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1------INFORMATION.
      WRITE(IOUT,1) TEXT,IBDCHN,KSTP,KPER
    1 FORMAT(1X,'"',4A4,'" BUDGET VALUES WILL BE SAVED ON UNIT',I3,
     1     ' AT END OF TIME STEP',I3,', STRESS PERIOD',I3)
C
      WRITE(IBDCHN) KSTP,KPER,TEXT,NCOL,NROW,NLAY
C
C2------WRITE AN UNFORMATTED RECORD CONTAINING VALUES FOR
C2------EACH CELL IN THE GRID. THE ARRAY IS DIMENSIONED
C2------(NCOL,NROW,NLAY)
      WRITE(IBDCHN) BUFF
C
C3------RETURN
      RETURN
      END
```

List of Variables for Module UBUDSV

Variation	Range	Definition
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
IBDCHN	Module	Unit number on which the array will be recorded.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
TEXT	Module	Label to be printed or recorded with the array data.

Narrative for Module ULASAV

Utility module ULASAV writes an unformatted record consisting of an array dimensioned (NCOL, NROW). The record containing the array is preceded by an unformatted record containing identifying information. The identifying information consists of:

KSTP	current time step	integer	1 word
KPER	current stress period	integer	1 word
PERTIM	elapsed time in the current stress period	real	1 word
TOTIM	elapsed time in the simulation	real	1 word
TEXT	label	character string	4 words
NCOL	number of columns	integer	1 word
NROW	number of rows	integer	1 word
ILAY	layer number	integer	1 word

```
      SUBROUTINE ULASAV(BUF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,
     1                  NROW,ILAY,ICHN)
C
C-----VERSION 1642 12MAY1987 ULASAV
C     ******************************************************************
C     SAVE 1 LAYER ARRAY ON DISK
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 TEXT
      DIMENSION BUF(NCOL,NROW),TEXT(4)
C     ------------------------------------------------------------------
C
C1------WRITE AN UNFORMATTED RECORD CONTAINING IDENTIFYING
C1------INFORMATION.
      WRITE(ICHN) KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY
C
C2------WRITE AN UNFORMATTED RECORD CONTAINING ARRAY VALUES
C2------THE ARRAY IS DIMENSIONED (NCOL,NROW)
      WRITE(ICHN) ((BUF(IC,IR),IC=1,NCOL),IR=1,NROW)
C
C3------RETURN
      RETURN
      END
```

List of Variables for Module ULASAV

Variable	Range	Definition
BUF	Module	Buffer containing data to be printed or recorded.
IC	Module	Index for columns.
ICHN	Module	Unit number on which the array is to be recorded.
ILAY	Module	Layer number.
IR	Module	Index for rows.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NROW	Global	Number of rows in the grid.
PERTIM	Package	Elapsed time during the current stress period.
TEXT	Module	Label to be printed or recorded with the array data.
TOTIM	Package	Elapsed time in the simulation.

Narrative for Module ULAPRS

Module ULAPRS prints a two-dimensional array in strip form (fig. 56) using one of twelve FORTRAN formats. Module ULAPRS performs its tasks in the following order:

1. Get the format code (IP). If it is less than 1 or greater than 12, set it equal to 12 (the default).

2. Use the format code (IP) to determine the number of values (NCAP) to be printed on one line.

3. Calculate the number of spaces used for each value (NCPF) and the number of strips (NSTRIP). Initialize the fields to store the first column (J1) and the last column (J2) for each strip.

4. Loop through the strips (DO STEPS 5-8).

5. Calculate the first (J1) and last (J2) column for this strip.

6. Print a title on each strip.

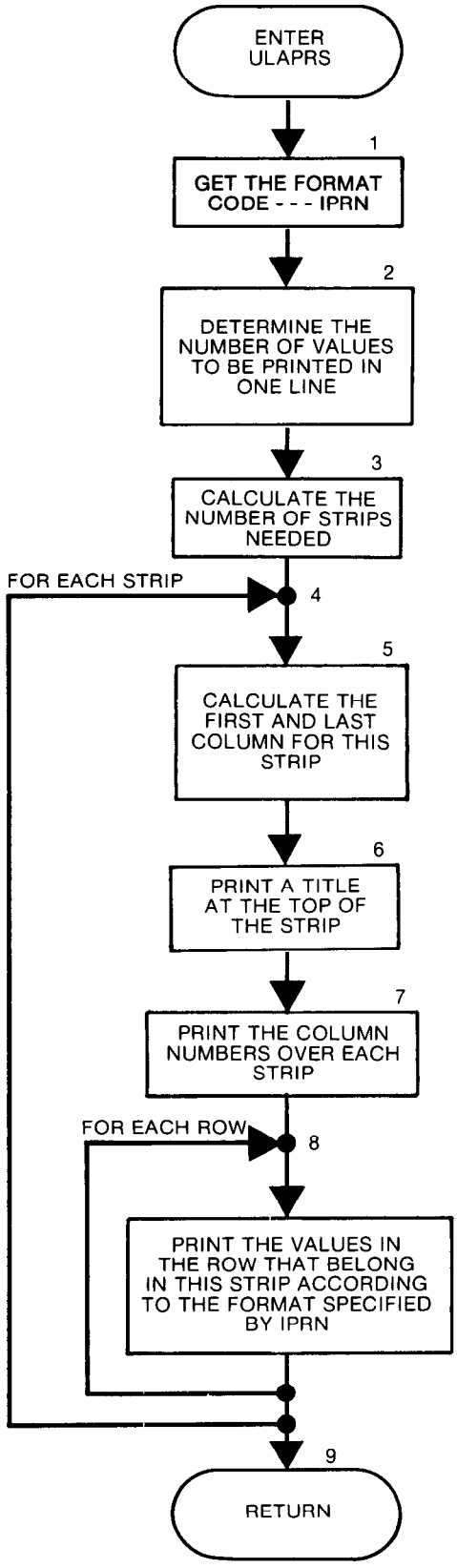
7. Call module UCOLNO to print the column numbers above each strip.

8. Loop through the rows printing columns J1 through J2 using the appropriate format (IP).

9. RETURN.

Flow Chart for Module ULAPRS

IPRN is a code indicating the format to be used in printing array values. If it is not between 1 and 12, it is set equal to 12.



```
      SUBROUTINE ULAPRS(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,IPRN,IOUT)
C
C
C-----VERSION 1640 12MAY1987 ULAPRS
C     ******************************************************************
C     PRINT A 1 LAYER ARRAY IN STRIPS
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 TEXT
      DIMENSION BUF(NCOL,NROW),TEXT(4)
C     ------------------------------------------------------------------
C
C1------MAKE SURE THE FORMAT CODE (IP OR IPRN) IS BETWEEN 1
C1------AND 12.
      IP=IPRN
      IF(IP.LT.1 .OR. IP.GT.12) IP=12
C
C2------DETERMINE THE NUMBER OF VALUES (NCAP) PRINTED ON ONE LINE.
      IF(IP.EQ.1) NCAP=11
      IF(IP.EQ.2) NCAP=9
      IF(IP.GT.2 .AND. IP.LT.7) NCAP=15
      IF(IP.GT.6 .AND. IP.LT.12) NCAP=20
      IF(IP.EQ.12) NCAP=10
C
C3------CALCULATE THE NUMBER OF STRIPS (NSTRIP).
      NCPF=129/NCAP
      ISP=0
      IF(NCAP.GT.12) ISP=3
      NSTRIP=(NCOL-1)/NCAP + 1
      J1=1-NCAP
      J2=0
C
C4------LOOP THROUGH THE STRIPS.
      DO 2000 N=1,NSTRIP
C
C5------CALCULATE THE FIRST(J1) & THE LAST(J2) COLUMNS FOR THIS STRIP
      J1=J1+NCAP
      J2=J2+NCAP
      IF(J2.GT.NCOL) J2=NCOL
C
C6-------PRINT TITLE ON EACH STRIP
      WRITE(IOUT,1) TEXT,ILAY,KSTP,KPER
    1 FORMAT(1H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
     1     ' IN STRESS PERIOD',I3/11X,71('-'))
C
C7------PRINT COLUMN NUMBERS ABOVE THE STRIP
      CALL UCOLNO(J1,J2,ISP,NCAP,NCPF,IOUT)
C
C8------LOOP THROUGH THE ROWS PRINTING COLS J1 THRU J2 WITH FORMAT IP
      DO 1000 I=1,NROW
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120), IP
C
C------------FORMAT 10G10.3
   10 WRITE(IOUT,11) I,(BUF(J,I),J=J1,J2)
   11 FORMAT(1H0,I3,2X,1PG10.3,10(1X,G10.3))
      GO TO 1000
C
C------------FORMAT 8G13.6
```

```
   20 WRITE(IOUT,21) I,(BUF(J,I),J=J1,J2)
   21 FORMAT(1H0,I3,2X,1PG13.6,8(1X,G13.6))
      GO TO 1000
C
C------------FORMAT 15F7.1
   30 WRITE(IOUT,31) I,(BUF(J,I),J=J1,J2)
   31 FORMAT(1H0,I3,1X,15(1X,F7.1))
      GO TO 1000
C
C------------FORMAT 15F7.2
   40 WRITE(IOUT,41) I,(BUF(J,I),J=J1,J2)
   41 FORMAT(1H0,I3,1X,15(1X,F7.2))
      GO TO 1000
C
C------------FORMAT 15F7.3
   50 WRITE(IOUT,51) I,(BUF(J,I),J=J1,J2)
   51 FORMAT(1H0,I3,1X,15(1X,F7.3))
      GO TO 1000
C
C------------FORMAT 15F7.4
   60 WRITE(IOUT,61) I,(BUF(J,I),J=J1,J2)
   61 FORMAT(1H0,I3,1X,15(1X,F7.4))
      GO TO 1000
C
C------------FORMAT 20F5.0
   70 WRITE(IOUT,71) I,(BUF(J,I),J=J1,J2)
   71 FORMAT(1H0,I3,1X,20(1X,F5.0))
      GO TO 1000
C
C------------FORMAT 20F5.1
   80 WRITE(IOUT,81) I,(BUF(J,I),J=J1,J2)
   81 FORMAT(1H0,I3,1X,20(1X,F5.1))
      GO TO 1000
C
C------------FORMAT 20F5.2
   90 WRITE(IOUT,91) I,(BUF(J,I),J=J1,J2)
   91 FORMAT(1H0,I3,1X,20(1X,F5.2))
      GO TO 1000
C
C------------FORMAT 20F5.3
  100 WRITE(IOUT,101) I,(BUF(J,I),J=J1,J2)
  101 FORMAT(1H0,I3,1X,20(1X,F5.3))
      GO TO 1000
C
C------------FORMAT 20F5.4
  110 WRITE(IOUT,111) I,(BUF(J,I),J=J1,J2)
  111 FORMAT(1H0,I3,1X,20(1X,F5.4))
      GO TO 1000
C
C------------FORMAT 9G11.4
  120 WRITE(IOUT,121) I,(BUF(J,I),J=J1,J2)
  121 FORMAT(1H0,I3,2X,1PG11.4,9(1X,G11.4))
C
 1000 CONTINUE
 2000 CONTINUE
C
C9------RETURN
      RETURN
      END
```

List of Variables for Module ULAPRS

Variable	Range	Definition
BUF	Module	Buffer containing data to be printed or recorded.
I	Module	Index for rows.
ILAY	Module	Layer number.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IP	Module	Format code.
IPRN	Module	Code for the format to be used when printing arrays.
ISP	Module	Number of spaces.
J	Module	Index for columns.
J1	Module	First column in a strip.
J2	Module	Last column in a strip.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
N	Module	Index for strips.
NCAP	Module	Number of columns on a line.
NCOL	Global	Number of columns in the grid.
NCPF	Module	Number of columns per field.
NROW	Global	Number of rows in the grid.
NSTRIP	Module	Number of strips.
TEXT	Module	Label to be printed or recorded with the array data.

Narrative for Module ULAPRW

Module ULAPRW prints a two-dimensional array in wrap form (fig. 56) using one of twelve FORTRAN formats. Module ULAPRW performs its tasks in the following order:

1. Print a header.

2. Set the format code (IP). If it is less than 1 or greater than 12, set it equal to 12 (the default).

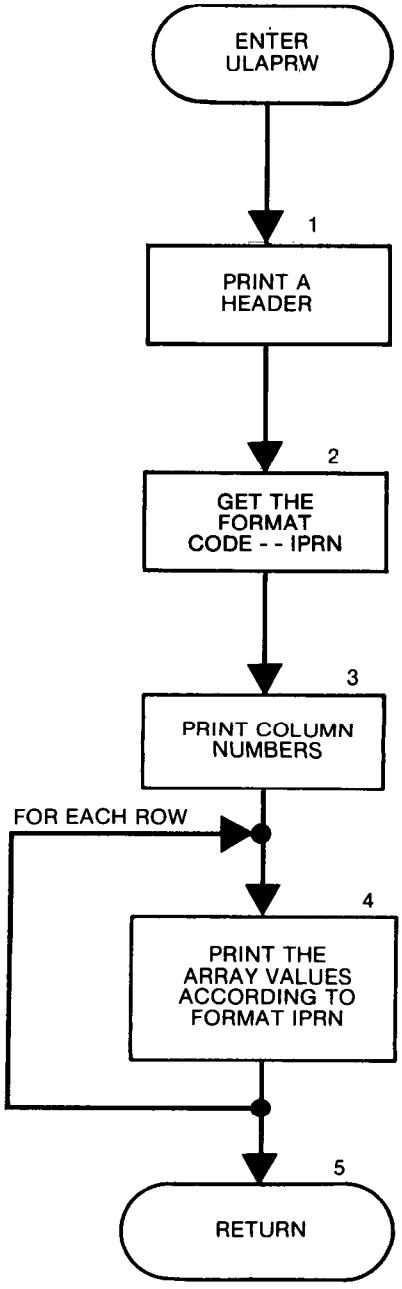
3. Call the module UCOLNO to print column numbers.

4. Loop through the rows printing each one in its entirety using the appropriate format code.

5. RETURN.

Flow Chart for Module ULAPRW

IPRN is a code indicating the format to be used in printing array values. If it is not between 1 and 12, it is set equal to 12.



```
      SUBROUTINE ULAPRW(BUF,TEXT,KSTP,KPER,NCOL,NROW,ILAY,IPRN,IOUT)
C
C
C-----VERSION 1642 12MAY1987 ULAPRW
C     ******************************************************************
C     PRINT 1 LAYER ARRAY
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 TEXT
      DIMENSION BUF(NCOL,NROW),TEXT(4)
C     ------------------------------------------------------------------
C
C1------PRINT A HEADER
      IF(ILAY.LE.0) GO TO 5
      WRITE(IOUT,1) TEXT,ILAY,KSTP,KPER
    1 FORMAT(1H1,10X,4A4,' IN LAYER',I3,' AT END OF TIME STEP',I3,
     1     ' IN STRESS PERIOD',I3/11X,71('-'))
C
C2------MAKE SURE THE FORMAT CODE (IP OR IPRN) IS
C2------BETWEEN 1 AND 12.
    5 IP=IPRN
      IF(IP.LT.1 .OR. IP.GT.12) IP=12
C
C3------CALL THE UTILITY MODULE UCOLNO TO PRINT COLUMN NUMBERS.
      IF(IP.EQ.1) CALL UCOLNO(1,NCOL,0,11,11,IOUT)
      IF(IP.EQ.2) CALL UCOLNO(1,NCOL,0,9,14,IOUT)
      IF(IP.GT.2 .AND. IP.LT.7) CALL UCOLNO(1,NCOL,3,15,8,IOUT)
      IF(IP.GT.6 .AND. IP.LT.12) CALL UCOLNO(1,NCOL,3,20,6,IOUT)
      IF(IP.EQ.12) CALL UCOLNO(1,NCOL,0,10,12,IOUT)
C
C4------LOOP THROUGH THE ROWS PRINTING EACH ONE IN ITS ENTIRETY.
      DO 1000 I=1,NROW
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120), IP
C
C------------ FORMAT 11G10.3
   10 WRITE(IOUT,11) I,(BUF(J,I),J=1,NCOL)
   11 FORMAT(1H0,I3,2X,1PG10.3,10(1X,G10.3)/(5X,11(1X,G10.3)))
      GO TO 1000
C
C------------ FORMAT 9G13.6
   20 WRITE(IOUT,21) I,(BUF(J,I),J=1,NCOL)
   21 FORMAT(1H0,I3,2X,1PG13.6,8(1X,G13.6)/(5X,9(1X,G13.6)))
      GO TO 1000
C
C------------ FORMAT 15F7.1
   30 WRITE(IOUT,31) I,(BUF(J,I),J=1,NCOL)
   31 FORMAT(1H0,I3,1X,15(1X,F7.1)/(5X,15(1X,F7.1)))
      GO TO 1000
```

```
C
C------------ FORMAT 15F7.2
   40 WRITE(IOUT,41) I,(BUF(J,I),J=1,NCOL)
   41 FORMAT(1H0,I3,1X,15(1X,F7.2)/(5X,15(1X,F7.2)))
      GO TO 1000
C
C------------ FORMAT 15F7.3
   50 WRITE(IOUT,51) I,(BUF(J,I),J=1,NCOL)
   51 FORMAT(1H0,I3,1X,15(1X,F7.3)/(5X,15(1X,F7.3)))
      GO TO 1000
C
C------------ FORMAT 15F7.4
   60 WRITE(IOUT,61) I,(BUF(J,I),J=1,NCOL)
   61 FORMAT(1H0,I3,1X,15(1X,F7.4)/(5X,15(1X,F7.4)))
      GO TO 1000
C
C------------ FORMAT 20F5.0
   70 WRITE(IOUT,71) I,(BUF(J,I),J=1,NCOL)
   71 FORMAT(1H0,I3,1X,20(1X,F5.0)/(5X,20(1X,F5.0)))
      GO TO 1000
C
C------------ FORMAT 20F5.1
   80 WRITE(IOUT,81) I,(BUF(J,I),J=1,NCOL)
   81 FORMAT(1H0,I3,1X,20(1X,F5.1)/(5X,20(1X,F5.1)))
      GO TO 1000
C
C------------ FORMAT 20F5.2
   90 WRITE(IOUT,91) I,(BUF(J,I),J=1,NCOL)
   91 FORMAT(1H0,I3,1X,20(1X,F5.2)/(5X,20(1X,F5.2)))
      GO TO 1000
C
C------------ FORMAT 20F5.3
  100 WRITE(IOUT,101) I,(BUF(J,I),J=1,NCOL)
  101 FORMAT(1H0,I3,1X,20(1X,F5.3)/(5X,20(1X,F5.3)))
      GO TO 1000
C
C------------ FORMAT 20F5.4
  110 WRITE(IOUT,111) I,(BUF(J,I),J=1,NCOL)
  111 FORMAT(1H0,I3,1X,20(1X,F5.4)/(5X,20(1X,F5.4)))
      GO TO 1000
C
C------------ FORMAT 10G11.4
  120 WRITE(IOUT,121) I,(BUF(J,I),J=1,NCOL)
  121 FORMAT(1H0,I3,2X,1PG11.4,9(1X,G11.4)/(5X,10(1X,G11.4)))
C
 1000 CONTINUE
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module ULAPRW

Variable	Range	Definition
BUF	Module	Buffer containing data to be printed or recorded.
I	Module	Index for rows.
ILAY	Module	Layer number.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IP	Module	Format code.
IPRN	Module	Code for the format to be used when printing arrays.
J	Module	Index for columns.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter. Reset at the start of each stress period.
NCOL	Global	Number of columns in the grid.
NROW	Global	Number of rows in the grid.
TEXT	Module	Label to be printed or recorded with the array data.

Narrative for Module UCOLNO

Module UCOLNO prints column numbers at the top of a page when arrays of numbers are printed. It performs its functions in the following order:

1. Calculate the number of columns to be printed (NLBL), the width of a line (NTOT), and the number of lines needed to print all of the column numbers (NWRAP). Initialize the fields J1 and J2 which contain the first and last column number on each print line.

2. Build and print each line (DO STEPS 3-6).

3. Clear the line buffer (BF) in which the line is built.

4. Determine the first (J1) and last (J2) column number for the current line.

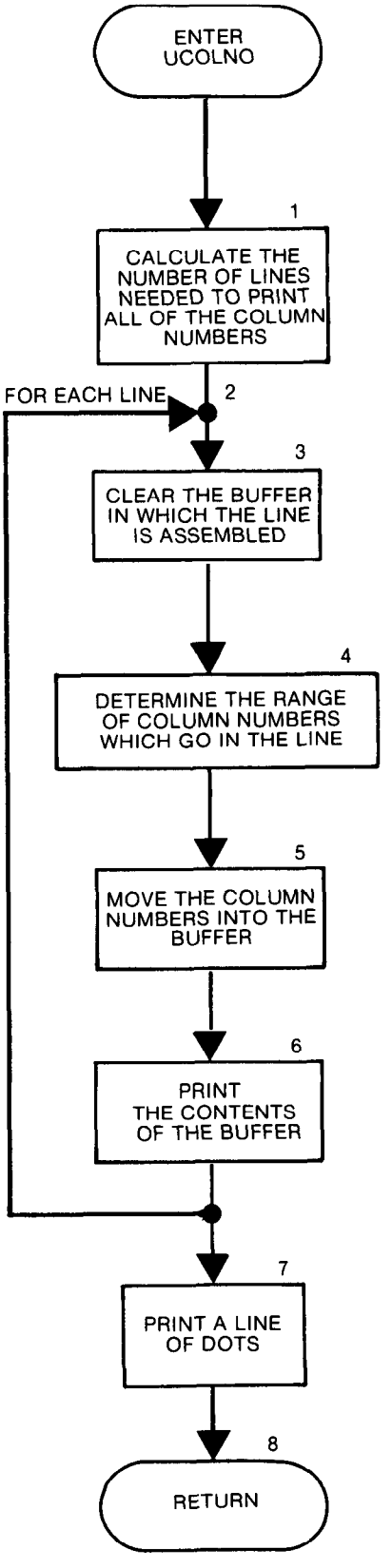
5. Put the column numbers in the line buffer. They are selected from the array DG. The indices I1, I2, and I3 point to the units digit, tens digit, and hundreds digit, respectively.

6. Print the line.

7. Print a line of dots.

8. RETURN.

Flow Chart for Module UCOLNO



```
      SUBROUTINE UCOLNO(NLBL1,NLBL2,NSPACE,NCPL,NDIG,IOUT)
C
C
C-----VERSION 1638 12MAY1987 UCOLNO
C     ******************************************************************
C     OUTPUT COLUMN NUMBERS ABOVE A MATRIX PRINTOUT
C        NLBL1 IS THE START COLUMN LABEL (NUMBER)
C        NLBL2 IS THE STOP COLUMN LABEL (NUMBER)
C        NSPACE IS NUMBER OF BLANK SPACES TO LEAVE AT START OF LINE
C        NCPL IS NUMBER OF COLUMN NUMBERS PER LINE
C        NDIG IS NUMBER OF CHARACTERS IN EACH COLUMN FIELD
C        IOUT IS OUTPUT CHANNEL
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 DOT,SPACE,DG,BF
      DIMENSION BF(130),DG(10)
C
      DATA DG(1),DG(2),DG(3),DG(4),DG(5),DG(6),DG(7),DG(8),DG(9),DG(10)/
     1         '0   ','1   ','2   ','3   ','4   ','5   ','6   ',
     2         '7   ','8   ','9   '/
      DATA DOT,SPACE/'.   ','    '/
C     ------------------------------------------------------------------
C
C1------CALCULATE # OF COLUMNS TO BE PRINTED (NLBL), WIDTH
C1------OF A LINE (NTOT), NUMBER OF LINES (NWRAP).
      WRITE(IOUT,1)
    1 FORMAT(1X)
      NLBL=NLBL2-NLBL1+1
      N=NLBL
      IF(NLBL.GT.NCPL) N=NCPL
      NTOT=NSPACE+N*NDIG
      IF(NTOT.GT.130) GO TO 50
      NWRAP=(NLBL-1)/NCPL + 1
      J1=NLBL1-NCPL
      J2=NLBL1-1
C
C2------BUILD AND PRINT EACH LINE
      DO 40 N=1,NWRAP
C
C3------CLEAR THE BUFFER (BF).
      DO 20 I=1,130
      BF(I)=SPACE
   20 CONTINUE
      NBF=NSPACE
C
C4------DETERMINE FIRST (J1) AND LAST (J2) COLUMN # FOR THIS LINE.
      J1=J1+NCPL
      J2=J2+NCPL
      IF(J2.GT.NLBL2) J2=NLBL2
C5------LOAD THE COLUMN #'S INTO THE BUFFER.
      DO 30 J=J1,J2
      NBF=NBF+NDIG
      I2=J/10
      I1=J-I2*10+1
      BF(NBF)=DG(I1)
      IF(I2.EQ.0) GO TO 30
      I3=I2/10
      I2=I2-I3*10+1
      BF(NBF-1)=DG(I2)
      IF(I3.EQ.0) GO TO 30
      BF(NBF-2)=DG(I3+1)
   30 CONTINUE
C
C6------PRINT THE CONTENTS OF THE BUFFER (I.E. PRINT THE LINE).
      WRITE(IOUT,31) (BF(I),I=1,NBF)
   31 FORMAT(1X,130A1)
C
   40 CONTINUE
C
C7------PRINT A LINE OF DOTS (FOR ESTHETIC PURPOSES ONLY).
   50 NTOT=NTOT+5
      IF(NTOT.GT.130) NTOT=130
      WRITE(IOUT,51) (DOT,I=1,NTOT)
   51 FORMAT(1X,130A1)
C
C8------RETURN
      RETURN
      END
```

List of Variables for Module UCOLNO

Variable	Range	Definition
BF	Module	DIMENSION (130), Buffer in which a line is assembled.
DG	Module	DIMENSION (10), Digits 0 through 9.
DOT	Module	Field containing a period.
I	Module	Index for BF.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
I1	Module	Index for DG (units digit).
I2	Module	Index for DG (tens digit).
I3	Module	Index for DG (hundreds digit).
J	Module	Index for column numbers.
J1	Module	First column number on the current line.
J2	Module	Last column number on the current line.
N	Module	Number of column numbers.
NBF	Module	Index for BF.
NCPL	Module	Number of column numbers per line.
NDIG	Module	Number of characters in each column number field.
NLBL	Module	Number of column numbers to be printed.
NLBL1	Module	Start column number.
NLBL2	Module	Stop column number.
NSPACE	Module	Number of blank spaces at start of line.
NTOT	Module	Total number of characters on a line.
NWRAP	Module	Number of lines needed in wrap format.
SPACE	Module	Field containing blanks.

Narrative for Module U2DREL

Module U2DREL reads values for a two-dimensional real array. First it reads an "array-control record." Then, based on the contents of the array-control record, it may read array values. The array-control record contains four fields: location (LOCAT), constant (CNSTNT), format (FMTIN), and printout indicator (IPRN). The LOCAT field determines where array values will come from. If LOCAT is positive, it is the unit number from which array values will be read in the format specified in FMTIN. If LOCAT is negative, the sign is reversed to give the unit number from which an unformatted record containing the array values will be read. (Before the array record is read, a record will be read and ignored. Thus output from the module ULASAV can be read.) If LOCAT is zero, all of the array values will be set equal to CNSTNT. When LOCAT is not zero and CNSTNT is not zero, the array values will be multiplied by the value of CNSTNT. The field IPRN contains a code number for a FORTRAN format to be used when printing the array.

Module U2DREL performs its tasks in the following order:

1. Read the array-control record (LOCAT, CNSTNT, FMTIN, and IPRN).

2. Use LOCAT to determine where the array values are coming from. GO TO STEPS 3, 4, OR 5.

3. If LOCAT equals zero, set all array values equal to CNSTNT, print a message to that effect, and RETURN.

4. If LOCAT is greater than zero, read array values according to the format in FMTIN. GO TO STEP 6.

5. If LOCAT is less than zero, read an unformatted dummy record and then read an unformatted record containing the array values. GO TO STEP 6.

6. If CNSTNT is not equal to zero, multiply array values by CNSTNT.

7. If IPRN is greater than or equal to zero, call utility module ULAPRW to print the array values using IPRN as the format code.

8. RETURN.

Flow Chart for Module U2DREL

Array Control Record controls the input of array values. It contains four fields: LOCAT, CNSTNT, FMTIN, and IPRN.

LOCAT is a code showing where array values will come from.

If LOCAT < 0, array values will be read from an unformatted record from a unit number equal to -LOCAT.

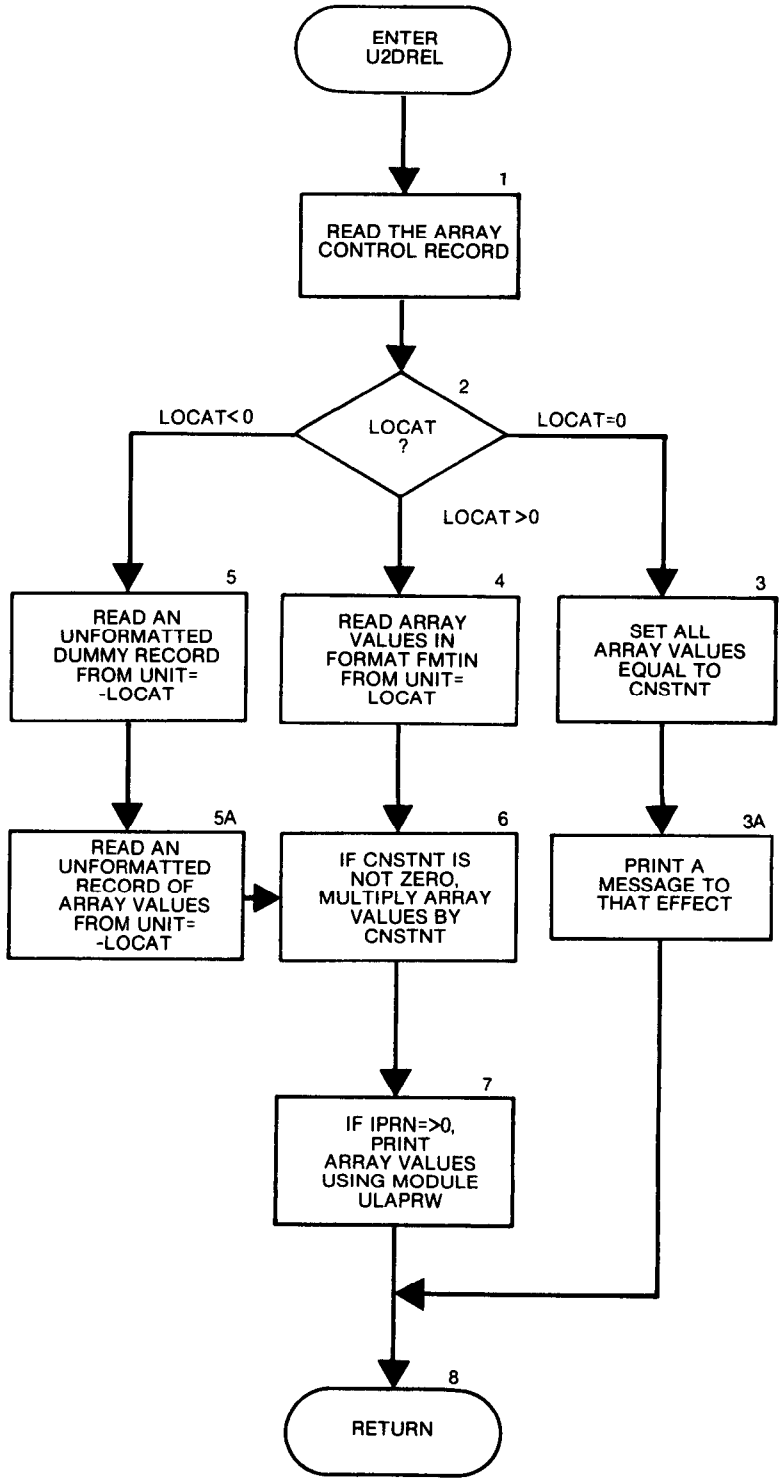
If LOCAT = 0, array values will be set equal to CNSTNT.

If LOCAT > 0, array values will be read from the unit number equal to LOCAT in the format specified in FMTIN.

CNSTNT is the constant to which all array values are set if LOCAT is equal to zero, and it is the constant by which all array values are multiplied if LOCAT is not equal to zero.

FMTIN is the format in which array values are read if LOCAT is greater than zero.

IPRN is a code showing the format to be used if array values are to be printed.



```
      SUBROUTINE U2DREL(A,ANAME,II,JJ,K,IN,IOUT)
C
C
C-----VERSION 1648 12MAY1987 U2DREL
C     *****************************************************************
C     ROUTINE TO INPUT 2-D REAL DATA MATRICES
C       A IS ARRAY TO INPUT
C       ANAME IS 24 CHARACTER DESCRIPTION OF A
C       II IS NO. OF ROWS
C       JJ IS NO. OF COLS
C       K IS LAYER NO. (USED WITH NAME TO TITLE PRINTOUT UNLESS K IS 0)
C       IN IS INPUT UNIT
C       IOUT IS OUTPUT UNIT
C     *****************************************************************
C
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      CHARACTER*4 ANAME
      CHARACTER*20 FMTIN
      DIMENSION A(JJ,II),ANAME(6)
C     -----------------------------------------------------------------
C
C1------READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT,CNSTNT,FMTIN,IPRN
    1 FORMAT(I10,F10.0,A20,I10)
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT) 200,50,90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT. RETURN
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 A(J,I)=CNSTNT
      IF(K.GT.0) WRITE(IOUT,2) ANAME,CNSTNT,K
    2 FORMAT(1H0,52X,6A4,' =',G15.7,' FOR LAYER',I3)
      IF(K.LE.0) WRITE(IOUT,3) ANAME,CNSTNT
    3 FORMAT(1H0,52X,6A4,' =',G15.7)
      RETURN
C
C4------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 IF(K.GT.0) WRITE(IOUT,4) ANAME,K,LOCAT,FMTIN
    4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
     1       I3,' USING FORMAT: ',A20/30X,96('-'))
      IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
    5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
     1       I3,' USING FORMAT: ',A20/30X,83('-'))
      DO 100 I=1,II
      READ (LOCAT,FMTIN) (A(J,I),J=1,JJ)
  100 CONTINUE
      GO TO 300
C
C5------LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY VALUES
  200 LOCAT=-LOCAT
      IF(K.GT.0) WRITE(IOUT,201) ANAME,K,LOCAT
  201 FORMAT(1H0,///30X,6A4,', LAYER',I3,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,73('-'))
      IF(K.LE.0) WRITE(IOUT,202) ANAME,LOCAT
  202 FORMAT(1H0,///30X,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,60('-'))
C
C5A-----READ AN UNFORMATTED DUMMY RECORD FIRST.
      READ(LOCAT)
      READ(LOCAT) A
C
C6------IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
  300 IF(CNSTNT.EQ.0.) GO TO 320
      DO 310 I=1,II
      DO 310 J=1,JJ
      A(J,I)=A(J,I)*CNSTNT
  310 CONTINUE
C
C7------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
  320 IF(IPRN.LT.0) RETURN
      CALL ULAPRW(A,ANAME,0,0,JJ,II,0,IPRN,IOUT)
      RETURN
C
C8------RETURN
      END
```

List of Variables for Module U2DREL

Variable	Range	Definition
A	Module	DIMENSION (JJ,II), Input array.
ANAME	Module	Label for printout of the input array.
CNSTNT	Module	Constant to which all array values are set if LOCAT is equal to zero or by which all array values are multiplied if LOCAT is not equal to zero.
FMTIN	Module	Format under which array values will be read.
I	Module	Index for rows.
II	Module	Number of rows.
IN	Module	Unit number from which the array control record will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRN	Module	Code for format to be used when printing the arrays.
J	Module	Index for columns.
JJ	Module	Number of columns.
K	Module	Layer number.
LOCAT	Module	Location of values to fill in the array. < 0, read an unformatted record containing the array values. = 0, set all the array values equal to constant (CNSTNT). > 0, read the formatted records containing the array values.

Narrative for Module U2DINT

Module U2DINT reads values for a two-dimensional integer array. First it reads an "array-control record." Then, based on the contents of the array-control record, it may read array values. The array-control record contains four fields: location (LOCAT), constant (ICONST), format (FMTIN), and printout indicator (IPRN). The LOCAT field determines where array values will come from. If LOCAT is positive, it is the unit number from which array values will be read in the format specified in FMTIN. If LOCAT is negative, the sign is reversed to give the unit number from which an unformatted record containing the array values will be read. (Before the array record is read, a record will be read and ignored. Thus output from the module ULASAV can be read.) If LOCAT is zero, all of the array values will be set equal to ICONST. When LOCAT is not zero and ICONST is not zero, the array values will be multiplied by the value of ICONST. The field IPRN (table 1) contains a code number for a FORTRAN format to be used when printing the array.

Module U2DINT performs its tasks in the following order:

1. Read the array-control record (LOCAT, ICONST, FMTIN, and IPRN).

2. Use LOCAT to determine where the array is coming from. GO TO STEPS 3, 4, OR 5.

3. If LOCAT equals zero, set all array values equal to CNSTNT, print a message to that effect, and RETURN.

4. If LOCAT is greater than zero, read array values according to the format in FMTIN. GO TO STEP 6.

5. If LOCAT is less than zero, read an unformatted dummy record and then read an unformatted record containing the array values. GO TO STEP 6.

6. If ICONST is not equal to zero, multiply array values by ICONST.

7. If IPRN is greater than or equal to zero, print the array values using IPRN as the format code.

8. Call utility module UCOLNO to print column numbers at the top of the page.

9. Print each row in the array.

10. Select the format for printing.

11. RETURN.

Flow Chart for Module U2DINT

Array Control Record controls the input of array values. It contains four fields: LOCAT, ICONST, FMTIN, and IPRN.

LOCAT is a code showing where array values will come from.

If LOCAT < 0, array values will be read from an unformatted record from a unit number equal to -LOCAT.

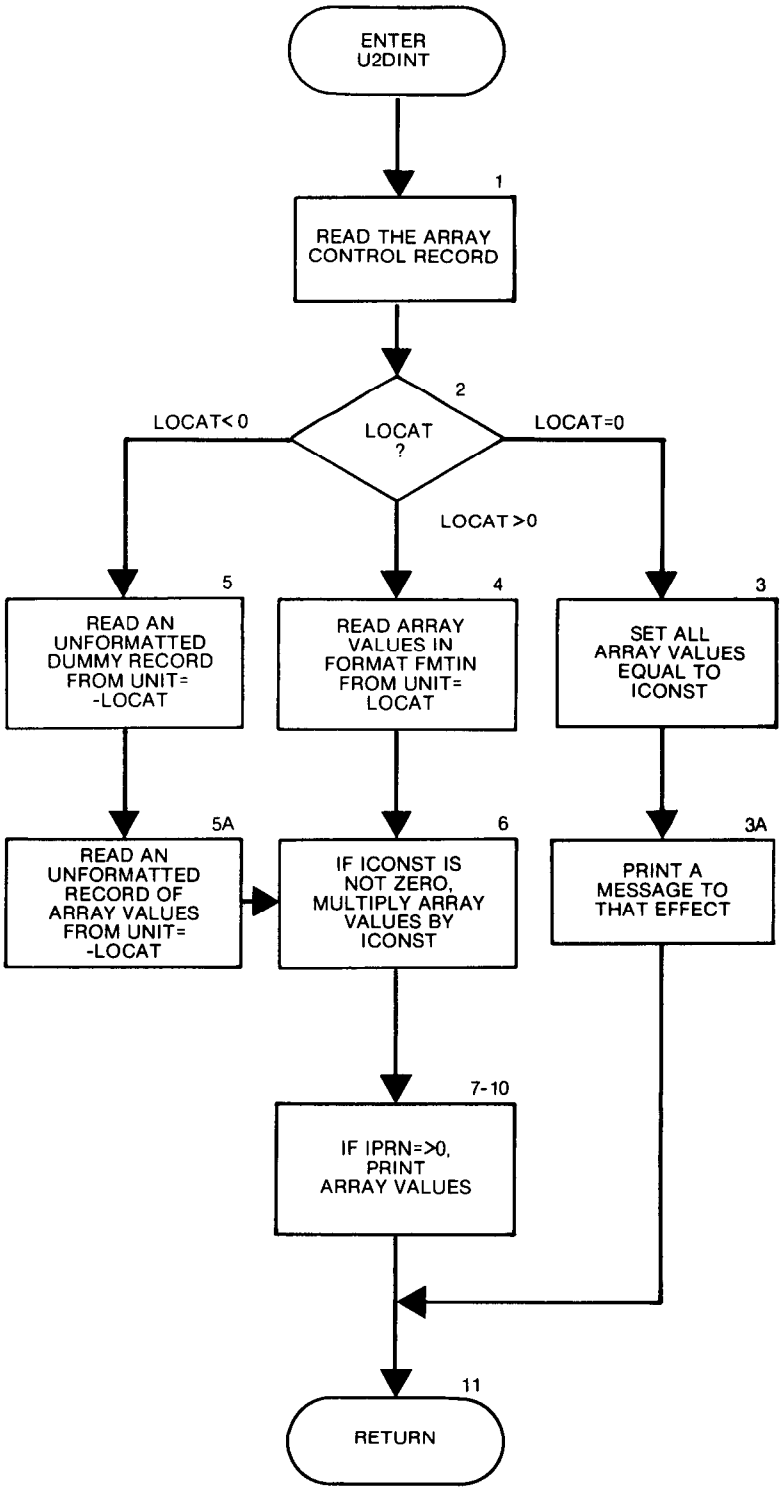
If LOCAT = 0, array values will be set equal to ICONST.

If LOCAT > 0, array values will be read from the unit number equal to LOCAT in the format specified in FMTIN.

ICONST is the constant to which all array values are set if LOCAT is equal to zero, and it is the constant by which all array values are multiplied if LOCAT is not equal to zero.

FMTIN is the format in which array values are read if LOCAT is greater than zero.

IPRN is a code showing the format to be used if array values are to be printed.



```
      SUBROUTINE U2DINT(IA,ANAME,II,JJ,K,IN,IOUT)
C
C
C-----VERSION 1645 12MAY1987 U2DINT
C     ******************************************************************
C     ROUTINE TO INPUT 2-D INTEGER DATA MATRICES
C       IA IS ARRAY TO INPUT
C       ANAME IS 24 CHARACTER DESCRIPTION OF IA
C       II IS NO. OF ROWS
C       JJ IS NO. OF COLS
C       K IS LAYER NO. (USED WITH NAME TO TITLE PRINTOUT UNLESS K IS 0)
C       IN IS INPUT UNIT
C       IOUT IS OUTPUT UNIT
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 ANAME
      CHARACTER*20 FMTIN
      DIMENSION IA(JJ,II),ANAME(6)
C     ------------------------------------------------------------------
C
C1------READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT,ICONST,FMTIN,IPRN
    1 FORMAT(I10,I10,A20,I10)
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT) 200,50,90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO ICONST. RETURN
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 IA(J,I)=ICONST
      IF(K.GT.0) WRITE(IOUT,2) ANAME,ICONST,K
    2 FORMAT(1H0,52X,6A4,' =',I15,' FOR LAYER',I3)
      IF(K.LE.0) WRITE(IOUT,3) ANAME,ICONST
    3 FORMAT(1H0,52X,6A4,' =',I15)
      RETURN
C
C4------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 IF(K.GT.0) WRITE(IOUT,4) ANAME,K,LOCAT,FMTIN
    4 FORMAT(1H0,///30X,6A4,' FOR LAYER',I3,' WILL BE READ ON UNIT',
     1       I3,' USING FORMAT: ',A20/30X,96('-'))
      IF(K.LE.0) WRITE(IOUT,5) ANAME,LOCAT,FMTIN
    5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',
     1        I3,' USING FORMAT: ',A20/30X,83('-'))
      DO 100 I=1,II
      READ (LOCAT,FMTIN) (IA(J,I),J=1,JJ)
  100 CONTINUE
      GO TO 300
C
C5------LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY VALUES
  200 LOCAT=-LOCAT
      IF(K.GT.0) WRITE(IOUT,201) ANAME,K,LOCAT
  201 FORMAT(1H0,///30X,6A4,', LAYER',I3,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,73('-'))
      IF(K.LE.0) WRITE(IOUT,202) ANAME,LOCAT
  202 FORMAT(1H0,///30X,6A4,
     1    ' WILL BE READ UNFORMATTED ON UNIT',I3/30X,60('-'))
C
```

```
C5A------READ AN UNFORMATTED DUMMY RECORD FIRST.
      READ(LOCAT)
      READ(LOCAT) IA
C
C6------IF ICONST NOT ZERO THEN MULTIPLY ARRAY VALUES BY ICONST.
  300 IF(ICONST.EQ.0) GO TO 320
      DO 310 I=1,II
      DO 310 J=1,JJ
      IA(J,I)=IA(J,I)*ICONST
  310 CONTINUE
C
C7------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
  320 IF(IPRN.LT.0) RETURN
      IF(IPRN.GT.5) IPRN=0
      IPRN=IPRN+1
C
C8------PRINT COLUMN NUMBERS AT TOP OF PAGE.
      IF(IPRN.EQ.1) CALL UCOLNO(1,JJ,0,10,12,IOUT)
      NL=125/IPRN/5*5
      IF(IPRN.GT.1) CALL UCOLNO(1,JJ,4,NL,IPRN,IOUT)
C
C9------PRINT EACH ROW IN THE ARRAY.
      DO 110 I=1,II
C
C10-----SELECT THE FORMAT
      GO TO(101,102,103,104,105,106), IPRN
C
C----------------FORMAT 10I11
  101 WRITE(IOUT,1001) I,(IA(J,I),J=1,JJ)
 1001 FORMAT(1H0,I3,2X,I11,9(1X,I11)/(5X,10(1X,I11)))
      GO TO 110
C
C----------------FORMAT 60I1
  102 WRITE(IOUT,1002) I,(IA(J,I),J=1,JJ)
 1002 FORMAT(1H0,I3,1X,60(1X,I1)/(5X,60(1X,I1)))
      GO TO 110
C
C----------------FORMAT 40I2
  103 WRITE(IOUT,1003) I,(IA(J,I),J=1,JJ)
 1003 FORMAT(1H0,I3,1X,40(1X,I2)/(5X,40(1X,I2)))
      GO TO 110
C
C----------------FORMAT 30I3
  104 WRITE(IOUT,1004) I,(IA(J,I),J=1,JJ)
 1004 FORMAT(1H0,I3,1X,30(1X,I3)/(5X,30(1X,I3)))
      GO TO 110
C
C----------------FORMAT 25I4
  105 WRITE(IOUT,1005) I,(IA(J,I),J=1,JJ)
 1005 FORMAT(1H0,I3,1X,25(1X,I4)/(5X,25(1X,I4)))
      GO TO 110
C
C----------------FORMAT 20I5
  106 WRITE(IOUT,1006) I,(IA(J,I),J=1,JJ)
 1006 FORMAT(1H0,I3,1X,20(1X,I5)/(5X,20(1X,I5)))
  110 CONTINUE
      RETURN
C
C11-----RETURN
      END
```

List of Variables for Module U2DINT

Variable	Range	Definition
ANAME	Module	Label for the printout of input array.
FMTIN	Module	Format under which the array values will be read.
I	Module	Index for rows.
IA	Module	DIMENSION (JJ,II), Input array.
ICONST	Module	Constant to which all array values are set if LOCAT is equal to zero or by which all array values are multiplied if LOCAT is not equal to zero.
II	Module	Number of rows.
IN	Module	Unit number from which the array-control record will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRN	Module	Code for format to be used when printing arrays.
J	Module	Index for columns.
JJ	Module	Number of columns.
K	Module	Layer number.
LOCAT	Module	Location of values to fill in the array. < 0, read an unformatted record containing the array values. = 0, set all the array values equal to constant (CNSTNT). > 0, read formatted records containing the array values.
NL	Module	Number of columns per line.

Narrative for Module U1DREL

Module U1DREL reads values for a one-dimensional real array. First it reads an "array-control record." Then, based on the contents of the array-control record, it may read array values. The array-control record contains four fields: location (LOCAT), constant (CNSTNT), format (FMTIN), and printout indicator (IPRN). The LOCAT field determines where array values will come from. If LOCAT is positive, it is the unit number from which array values will be read in the format specified in FMTIN. If LOCAT is zero, all of the array values will be set equal to CNSTNT. If LOCAT is not zero and CNSTNT is not zero, the array values will be multiplied by the value of CNSTNT. The field IPRN (table 2) contains a code number for a FORTRAN format to be used when printing the array.

Module U1DREL performs its tasks in the following order:

1. Read the array-control record (LOCAT, CNSTNT, FMTIN, and IPRN).

2. Use LOCAT to determine where the array is coming from (DO STEPS 3 OR 4).

3. If LOCAT equals zero, set all array values equal to CNSTNT and print a message to that effect. RETURN.

4. If LOCAT is greater than zero, read array values according to the format in FMTIN.

5. If CNSTNT is not equal to zero, multiply the array values by CNSTNT.

6. If IPRN is greater than or equal to zero, print the array values.

7. RETURN.

Flow Chart for Module U1DREL

Array Control Record controls the input of array values. It contains four fields: LOCAT, CNSTNT, FMTIN, and IPRN.

LOCAT is a code showing where array values will come from.

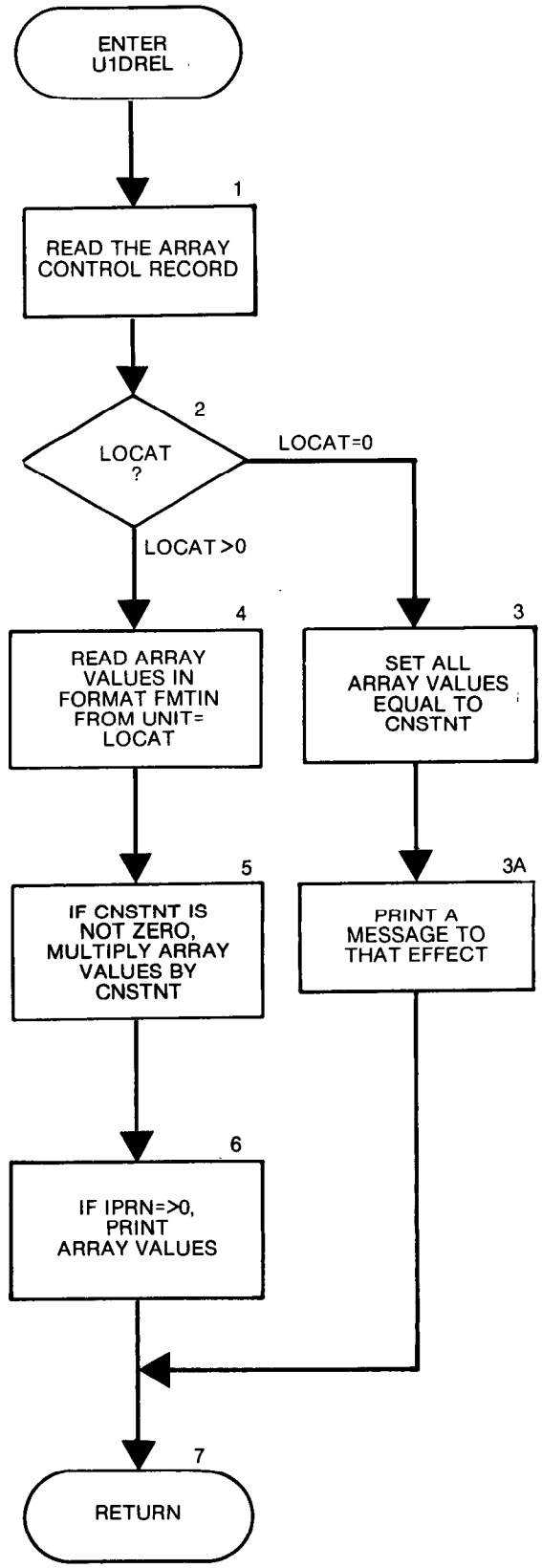
If LOCAT = 0, array values will be set equal to CNSTNT.

If LOCAT > 0, array values will be read from the unit number equal to LOCAT in the format specified in FMTIN.

CNSTNT is the constant to which all array values are set if LOCAT is equal to zero, and it is the constant by which all array values are multiplied if LOCAT is not equal to zero.

FMTIN is the format in which array values are read if LOCAT is greater than zero.

IPRN is a code showing the format to be used if array values are to be printed.



```
      SUBROUTINE U1DREL(A,ANAME,JJ,IN,IOUT)
C
C
C-----VERSION 1643 12MAY1987 U1DREL
C     ******************************************************************
C     ROUTINE TO INPUT 1-D REAL DATA MATRICES
C       A IS ARRAY TO INPUT
C       ANAME IS 24 CHARACTER DESCRIPTION OF A
C       JJ IS NO. OF ELEMENTS
C       IN IS INPUT UNIT
C       IOUT IS OUTPUT UNIT
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 ANAME
      CHARACTER*20 FMTIN
      DIMENSION A(JJ),ANAME(6)
C     ------------------------------------------------------------------
C
C1------READ ARRAY CONTROL RECORD.
      READ (IN,1) LOCAT,CNSTNT,FMTIN,IPRN
    1 FORMAT(I10,F10.0,A20,I10)
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM.
      IF(LOCAT.GT.0) GO TO 90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT. RETURN
      DO 80 J=1,JJ
   80 A(J)=CNSTNT
      WRITE(IOUT,3) ANAME,CNSTNT
    3 FORMAT(1H0,52X,6A4,' =',G15.7)
      RETURN
C
C4------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN.
   90 WRITE(IOUT,5) ANAME,LOCAT,FMTIN
    5 FORMAT(1H0,///30X,6A4,' WILL BE READ ON UNIT',I3,
     1         ' USING FORMAT: ',A20/30X,79('-')/)
      READ (LOCAT,FMTIN) (A(J),J=1,JJ)
C
C5------IF CNSTNT NOT ZERO THEN MULTIPLY ARRAY VALUES BY CNSTNT.
      IF(CNSTNT.EQ.0.) GO TO 120
      DO 100 J=1,JJ
  100 A(J)=A(J)*CNSTNT
C
C6------IF PRINT CODE (IPRN) =>0 THEN PRINT ARRAY VALUES.
 120  IF(IPRN.LT.0) RETURN
      WRITE(IOUT,1001) (A(J),J=1,JJ)
 1001 FORMAT((1X,1PG12.5,9(1X,G12.5)))
      RETURN
C
C7------CONTINUE
      END
```

List of Variables for Module U1DREL

Variable	Range	Definition
A	Module	DIMENSION (JJ), Input array.
ANAME	Module	Label for printout of the input array.
CNSTNT	Module	Constant to which all array values are set if LOCAT is equal to zero or by which all array values are multiplied if LOCAT is not equal to zero.
FMTIN	Module	Format under which the array values will be read.
IN	Module	Unit number from which the array control record will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPRN	Module	Code for the format to be used when printing the arrays.
J	Module	Array index.
JJ	Module	Number of elements in the array.
LOCAT	Module	Location of values to fill in the array. < 0, read an unformatted record containing the array values. = 0, set all the array values equal to constant (CNSTNT). > 0, read the formatted records containing the array values.

REFERENCES

Collins, R. E., 1961, Flow of fluids through porous materials; New York, Reinhold Publishing Corp., 270 p.

Crichlow, Henry B., 1977, Modern reservoir engineering - A simulation approach; Englewood Cliffs, N.J., Prentice Hall Inc., 354 p.

McDonald, M.G., and Harbaugh, A.W., 1984, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 83-875 , 528 p.

Peaceman, Donald W., 1977, Fundamentals of numerical reservoir simulation; New York, Elsevier Scientific Publishing Company, 176 p.

Remson, Irwin, Hornberger, George M. and Molz, Fred J., 1971, Numerical methods in subsurface hydrology; New York, Wiley-Interscience, 389 p.

Rushton, K. R., Redshaw S. C., 1979, Seepage and groundwater flow-numerical analysis by analog and digital methods; New York, John Wiley and Sons

Trescott, Peter C., 1975, Documentation of finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Open-File Report 75-438, 32 p.

Trescott, Peter C. and Larson, S. P., 1976, Supplement to Open-File Report 75-438, Documentation of finite-difference model of three-dimensional ground-water flow; U.S. Geological Survey Open-File Report 76-591, 21 p.

Trescott, P. C., Pinder, G. F., and Larson, S. P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 7, Chapter C1, 116 p.

Weinstein, H. C., Stone, H. L., and Kwan, T. V., 1969, Iterative procedure for solution of systems of parabolic and elliptic equations in three dimensions: Indus. Engineering Chemistry Fundamentals, v. 8, no. 2, p. 281-287.

APPENDIX A

PROGRAM PORTABILITY

Introduction

One of the major design requirements for the model program was that it should be portable. A portable program is one that can be run with a minimum of modification on most computers that are physically capable of running a program of its type. The goal of portability for the model program has been attained as evidenced by the fact that it has run successfully in either its present form or the earlier form (McDonald and Harbaugh, 1984) on computers manufactured by many companies including mainframe computers, minicomputers, and microcomputers. The following discussion explains in more detail the concept of portability, what was done to maximize portability of the model program, and circumstances that might require that the program be changed in order to run successfully on a particular computer.

The Impact of the Programming Language on Portability

The programming language is the most important factor that determines program portability. There are a variety of programming languages available, and for each language, there are numerous versions which have resulted from the desire of vendors to improve the power of the language and to take advantage of the hardware features of their particular computers. The most commonly available language suitable to use for the model program is FORTRAN. There are two versions defined by the American National Standards Institute (ANSI) on which most commercial versions are based, ANSI X3.9-1966 and ANSI X3.9-1978.[1] These versions are commonly referred to as FORTRAN 66 and FORTRAN 77, respectively. FORTRAN 66 was selected for the original version of the model because, at the time, it was far more widely supported than was FORTRAN 77. Now that the FORTRAN 77 standard is more widely supported, the program has been converted to this standard.

The program in this report is nearly identical to the original program except for the changes required to make it comply with the FORTRAN 77 standard. These changes are minor and are described by McDonald and Harbaugh (1984, p. 505). The conversion was done solely for the purpose of making the program comply with the FORTRAN 77 standard; many of the features that make FORTRAN 77 more powerful than FORTRAN 66 were not used. An effort to make more extensive FORTRAN 77 revisions to the program is judged to be uneconomical. Such an effort would not result in significant improvements in program clarity or efficiency.

---

[1] American National Standards Institute, 1966, FORTRAN: American National Standards Institute, X3.9-1966, 36 p.

American National Standards Institute, 1978, Programming language FORTRAN: American National Standards Institute, X3.9-1978, chs. 1-18.

Most commercial versions of FORTRAN 77 compilers include some extended features not defined as part of the FORTRAN 77 standard. Such features vary widely among computer vendors and were not used in the model program. The program contains only one exception to complete compliance to the FORTRAN 77 standard that the authors are aware of. This exception, which is explained in a following section (see The Impact of Allocating Array Storage in a Single Array on Portability), is commonly allowed on computers and should not be a major restriction to portability. Because the program closely follows the FORTRAN 77 standard, the program should work on any computer supporting this language provided that the computer has adequate computational power.

It is recognized that some users may want the program converted back into the FORTRAN 66 language because they have access only to older compilers. Only a few changes are required to convert the program back to FORTRAN 66. Information about how to convert is provided in a following section (see Conversion to FORTRAN 66).

The Impact of Computational Precision on Portability

Variation of precision among computers causes some problems with program portability. Computational precision refers to the accuracy at which numbers are calculated and stored in the computer. To prevent the imposition of constraints on the computers on which FORTRAN is implemented, the computational precision was not defined as part of the standards. The accuracy of model results are dependent on the computational precision, so precision must be considered when moving the model program among computers.

The model program was developed on computers using 32 binary bits to represent single precision real numbers. This gives from 6 to 7 decimal digits of precision and includes values that range in magnitude from approximately 10**-39 to 10**38. Double precision real numbers are represented by 64 binary bits and range in magnitude from approximately 10**-10000 to 10**10000 with 14 to 15 decimal digits of precision. The head array, HNEW, and some variables in the solvers are stored as double precision, and accordingly many calculations in the solvers are double precision. This was necessary for accuracy under some conditions. The model program should perform adequately for most problems on computers that use 32 bits for single precision and 64 bits for double precision. However, the required precision depends on the problem being simulated. Thus, the user must ultimately determine if adequate precision is being used for solving a particular problem.

There are some situations for which there is a need to modify the program to make all real number calculations in double precision. If using a computer that represents single precision real numbers with less than 32 bits, then double precision is probably necessary for all real numbers and calculations. Even on computers that represent single precision numbers with 32 bits, certain problems are difficult to solve without making all real numbers and calculations double precision. Unfortunately, it is difficult to predict if a specific problem requires all double precision. Simulations with very large numbers of cells, for example more than 50000, are more likely to have precision problems than are smaller simulations. Simulations in which there are areas having significant ground-water flow and yet the heads in the adjacent model cells in these areas are equal within .01 percent or less are

also more likely to have precision problems. In addition, precision problems depend on the preciseness of the attempted solution. In general, the symptoms of inadequate precision are either a poor volumetric flow balance or lack of convergence by the solver. However, these same symptoms are more commonly caused by bad input data or improper adjustment of parameters that control iteration. Because precision problems are in general fairly unlikely and use of all double precision results in increased memory and computer time usage, conversion to double precision should probably be done as a last resort in order to solve convergence problems. If the program is converted to all double precision, almost twice the computer memory is required for model data. To convert all real numbers and calculations to double precision, do the following:

1. Declare all single precision real arrays and variables as DOUBLE PRECISION in the main program and in every subroutine. This can be done by adding the statement

```
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

   to the beginning of the specifications section of every subroutine and the main program. Alternately, some compilers have a special command that will accomplish this without the need to modify the program itself.

2. In the Basic Package Allocate module (subroutine BAS1AL) change the statement

```
ISUM=ISUM+2*NRCL
```

   to

```
ISUM=ISUM+NRCL
```

3. Change the real intrinsic functions in subroutine SSIP1I to their double precision equivalents. Specifically, change all occurrences of AMAX1 to DMAX1, and AMIN1 to DMIN1.

If using a computer on which significantly more than 32 bits are used to represent single precision real numbers, then the partial use of double precision data and calculations as included in the present model program may be unnecessary. The program should still work on such computers without modification, but computational time and memory would be saved if double precision were eliminated. The program was designed so that changing to all single precision would be easy. For this reason, the use of double precision intrinsic functions and constants is avoided, and all conversions between real and double precision are done by implied type changes in assignment statements. If it is determined that the use of all single precision is acceptable, make the following changes to the program:

1. Delete all DOUBLE PRECISION specification statements.

2. In the Basic Package Allocate module (subroutine BAS1AL) change the statement

```
ISUM=ISUM+2*NRCL
```

   to

```
ISUM=ISUM+NRCL
```

The model program was developed on computers that use 32 bits for integer numbers and calculations. This is more than adequate to represent the range of integer numbers that are used in the program under any conditions. The largest integer that the computer must be able to represent is the dimensioned size of the X array in the main program. Many FORTRAN compilers allow one to specify that integers be represented by 16 bits, which does not provide enough precision for larger model simulations. That is, a 16 bit integer in a typical computer can represent numbers in the range -32768 to 32767, and an X array dimension of 32767 is adequate only for simulations that have 2200 to 3000 cells, depending on what options are used.

The Impact of Allocating Array Storage in a Single Array on Portability

Because almost all model data are stored in the X array, which is dimensioned in the main program, this array can be quite large. It is generally 10 to 15 times the size of the number of model cells. Some computers require that special compiler options be used when an array exceeds a specified size. Users are cautioned to be aware of this and use the appropriate options as needed. The result of using the incorrect option can sometimes be that the program will execute without producing error messages, but answers will be incorrect.

All computers limit the amount of array space that a program can use. If a user's simulation exceeds this limit, the only options may be to use a different computer or make the simulation smaller. However, some computers make a distinction between total amount of memory used for all arrays and the total used by a single array. That is, some computers might allow 4 arrays each having 32000 elements, but not allow a single array consisting of 128000 elements. On such a computer, it is quite possible that one could exceed the size limit for a single array without exceeding the total array size limit because nearly all model data are stored in the single X array. If this situation occurs, it is possible to break the X array into smaller pieces. Although this can be done by modifying only the main program, the modification is fairly complex. Such modification requires a knowledgeable programmer who has a clear understanding of how model data are stored within the X array. Before making such a modification, it would be prudent to assess how long the desired simulation might take to execute. Generally, computer execution speed is more of a constraint on maximum problem size than is array size.

The only known exception to the use of the FORTRAN 77 standard in the model program is that the data type of actual subroutine arguments does not always match the type of the corresponding dummy arguments. The standard requires the data type of actual and dummy arguments to match. This only happens in the main program where it calls subroutines using actual arguments that are elements from the X array. The X array itself is data type real, and several model arrays stored withi X are either integer (arrays IOFLG, IBOUND, IRCH, IEVT, and LRCH) or double precision (array HNEW). For example, actual argument X(LCIRCH) in the main program is passed to dummy argument IRCH in subroutine RCH1RP. X(LCIRCH) is of type real, and IRCH is of type integer. This practice has not been a problem in the past. Should this become a problem on future computers, required changes will be fairly minor, affecting only the main program and some of the Allocate modules.

The Impact on Portability of Preconnected File Units

FORTRAN 77 provides 2 ways for a file unit to become connected to (associated with) a file -- preconnection and the OPEN statement. The model program use preconnection. This means that the computer's operating system provides the connection between files and file units prior to program execution. Often the user must issue commands to the system, which connect the necessary files and file units, prior to running the program. The specific method for doing this varies among computers. Generally, preconnection is adequate, but it may be inconvenient on some computers. Modification of the main program to use OPEN statements is a simple task for a programmer.

Conversion to FORTRAN 66

Because the model program uses only one feature of FORTRAN 77 that is not part of FORTRAN 66, few changes are needed in order to make the program comply with the FORTRAN 66 standard. All of the required changes are a result of differences in the way character (alphanumeric) data are handled by the two versions of FORTRAN. Any variables or arrays holding character data must be declared to be the character data type in a FORTRAN 77 program; in a FORTRAN 66 program, character data are stored in numeric variables or arrays. The specific changes that are required to make the model program comply with the FORTRAN 66 standard are shown below. It is assumed that at least four characters can be stored in a single precision real variable or array element.

1. Delete all CHARACTER*4 statements throughout all subroutines and the main program.

2. In each of the array reading utility modules (U1DREL, U2DREL, and U2DINT), change the statement

```
      CHARACTER*20 FMTIN
```

   to

```
      DIMENSION FMTIN(5)
```

3. In each of the array reading utility modules (U1DREL, U2DREL, and U2DINT), change all occurrences of "A20" to "5A4". For example, change

```
     1        ' USING FORMAT: ',A20/30X,79('-')/)
```

   in subroutine U1DREL to

```
     1        ' USING FORMAT: ',5A4/30X,79('-')/)
```

4. To make the program strictly comply with the FORTRAN 66 standard, it is necessary to change character constants in DATA statements to Hollerith constants. However, most FORTRAN 66 compilers accept character constants specified using FORTRAN 77 notation (using apostrophes). Thus, it is unlikely that this change will be required.

APPENDIX B

SPACE REQUIREMENTS IN THE X ARRAY

The outline below gives the X-array space requirements for each package. The formulas can be used to calculate the exact size of the X array for a given problem. However, it is generally easier to simply run the model program and let it calculate the size. The total space required is printed as part of the model printout even if the X array is dimensioned too small. As a rough estimate, the X array is approximately 10 to 15 times the number of nodes in the model depending on the options selected. In the outline below, NODES is defined as NCOL*NROW*NLAY, the number of nodes in the model.

I. BAS Package

A. 8*NODES + (NLAY - 1)*NCOL*NROW + NROW + NCOL + 4*NLAY

B. Additionally, add NODES if start head is saved (ISTRT is not 0), and add NODES if BUFF is separate from RHS (IAPART is not 0)

II. BCF Package

A. NLAY

B. If a transient simulation (ISS is 0), add NODES

C. For each layer where LAYCON is one or three, add 2*NCOL*NROW

D. For each layer where LAYCON is two or three, add NCOL*NROW

E. If a transient simulation (ISS is 0), for each layer where LAYCON is two or three, add NCOL*NROW

III. WEL Package -- 4*MXWELL

IV. DRN Package -- 5*MXDRAN

V. RIV Package -- 6*MXRIVR

VI. EVT Package

A. Option 1 -- 3*NCOL*NROW

B. Option 2 -- 4*NCOL*NROW

VII. GHB Package -- 5*MXBND

VIII. RCH Package

A. Options 1 and 3 -- NCOL*NROW

B. Option 2 -- 2*NCOL*NROW

IX. SIP Package -- 4*NODES + 4*MXITER + NPARM

X. SOR Package -- (NLAY + 4)*NCOL*NLAY + 4*MXITER

Generally, it is advisable to have the X-array dimension relatively close to the amount of space needed for a specific problem. On the other hand, it is inconvenient to redimension the X array every time a new option is selected. Several load modules with the size of the X array differing by a factor of two should be adequate.

APPENDIX C

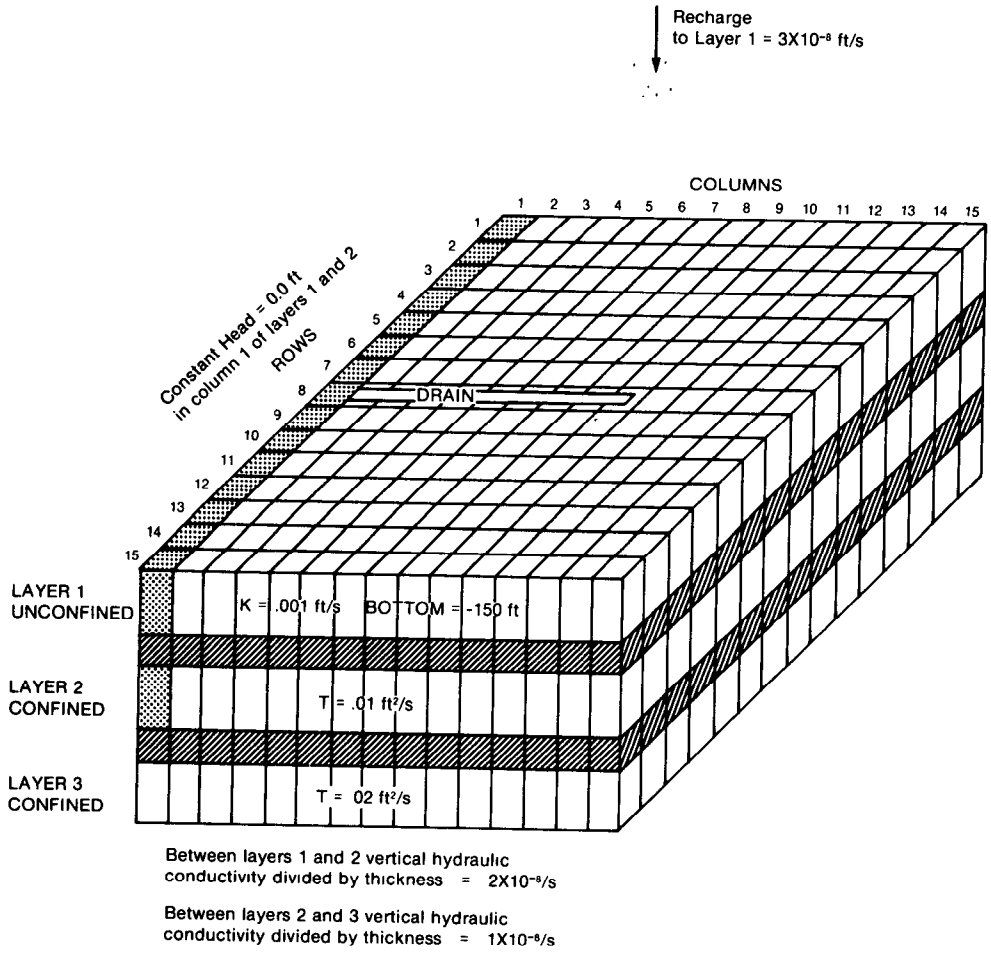
CONTINUATION OF A PREVIOUS RUN

There is often value in breaking long simulations into several short model runs. This allows one to decide, between runs, whether or not to continue the simulation. Although the model program in this report does not have a special option for making continuation runs, it is quite simple to continue a simulation by using the output of one run as input of the next. Simply save the heads from the run that is to be continued on a disk file, and specify that file as starting heads for the next run. The subroutine that reads starting heads (U2DREL) is capable of reading model-generated disk files of saved heads without the need for reformatting. Because volumetric budget terms are always set to zero at the start of a model run, the printed budget on a model run represents only that one run, not the total of all runs in a series of continuation runs. If a total budget for a series of continuation runs is desired, the totals from each run can be added externally. Similarly, the model program keeps track of simulation time only for single model runs, but total simulation time for a series of continuation runs can be calculated externally by adding the simulation times of each run.

APPENDIX D

SAMPLE PROBLEM

This sample problem is intended to illustrate input and output from the program. There are three simulated layers, as shown in the accompanying illustration, which are separated from each other by confining layers. Each layer is a square 75,000 feet on a side and is divided by a grid into 15 rows and 15 columns which form squares 5,000 feet on a side. Flow within the confining layers is not simulated, but the effects of the confining layers on flow between the active layers are incorporated in the vertical leakance (Vcont) terms. Flow into the system is infiltration from precipitation; flow out of the system is to buried drain tubes, discharging wells, and a lake which is represented by a constant-head boundary.



Setting starting heads equal to 0.0, the program was run to get a steady-state solution. The Strongly Implicit Procedure was used to solve the system of difference equations: the error criterion was set at 0.001 feet, the acceleration parameter was set to 1.0, and the maximum number of iterations was set equal to 50. A seed of 0.001 was specified for use in calculating the iteration parameters; 31 iterations were needed to close.

List of Wells

$Q = 5\ ft^3/s$ for each well

Layer	Row	Column
3	5	11
2	4	6
2	6	12
1	9	8
1	9	10
1	9	12
1	9	14
1	11	8
1	11	10
1	11	12
1	11	14
1	13	8
1	13	10
1	13	12
1	13	14

List of Drains

Conductance = $1\ ft^2/s$

Layer	Row	Column	Elevation
1	8	2	0.0
1	8	3	0.0
1	8	4	10.0
1	8	5	20.0
1	8	6	30.0
1	8	7	50.0
1	8	8	70.0
1	8	9	90.0
1	8	10	100.0

Input Data for Sample Problem

Note that input item names are included in many of the data records shown below. These are included for identification purposes. These names are ignored by the model program when it reads the data because they are placed in columns beyond those used to hold model data.

Input for FORTRAN unit 1 -- Basic Package:

```
SAMPLE----3 LAYERS, 15 ROWS, 15 COLUMNS; STEADY STATE; CONSTANT HEADS COLUMN 1,
LAYERS 1 AND 2; RECHARGE, WELLS AND DRAINS
         3        15        15         1         1
 11 12 13  0  0  0  0 18 19  0  0 00
         0         0     IAPART,ISTRT
         1         1(15I3)                       3    IBOUND-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  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  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(15I3)                       3    IBOUND-2
 -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  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  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
         0         1                                  IBOUND-3
    999.99
         0        0.                                  HEAD-1
         0        0.                                  HEAD-2
         0        0.                                  HEAD-3
    86400.         1        1.      PERLEN,NSTP,TSMULT
```

```
Input for FORTRAN unit 11 -- Block-Centered Flow Package:
        1         0      ISS,IBCFBD
 1 0 0
        0        1.                                          TRPY
        0     5000.                                          DELR
        0     5000.                                          DELC
        0     .001                                           HY-1
        0     -150.                                          BOT-1
        0     2.E-8                                          VHY/THICK-1
        0       .01                                          T-2
        0     1.E-8                                          VHY/THICK-2
        0       .02                                          T-3

Input for FORTRAN unit 18 -- Recharge Package:
        1         0    NRCHOP,IRCHBD
        1              INRECH
        0     3.E-8                                          RECH-1

Input for FORTRAN unit 19 -- Strongly Implicit Procedure Package:
       50         5       MXITER,NPARM
       1.      .001         0      .001         1 ACCL,ERR,IPCALC,WSEED

Input for FORTRAN unit 13 -- Drain Package:
        9         0    MXDRAI,IDRNBD
        9              NDRAIN
        1         8         2        0.     1.E00
        1         8         3        0.     1.E00
        1         8         4       10.     1.E00
        1         8         5       20.     1.E00
        1         8         6       30.     1.E00
        1         8         7       50.     1.E00
        1         8         8       70.     1.E00
        1         8         9       90.     1.E00
        1         8        10      100.     1.E00

Input for FORTRAN unit 12 -- Well Package:
       15         0    MXWELL,IWELBD
       15              ITMP (NWELLS)
        3         5        11      -5.
        2         4         6      -5.
        2         6        12      -5.
        1         9         8      -5.
        1         9        10      -5.
        1         9        12      -5.
        1         9        14      -5.
        1        11         8      -5.
        1        11        10      -5.
        1        11        12      -5.
        1        11        14      -5.
        1        13         8      -5.
        1        13        10      -5.
        1        13        12      -5.
        1        13        14      -5.
```

Sample Problem Output

Note that it should not be expected that the outputs from running the same problem on different computers will exactly match. Small variations in output can be caused by differences in the way real numbers are stored and calculated. Storage and calculation of real numbers depend on the specific computer hardware, the FORTRAN compiler, and the math library that is loaded with the compiled program. Output variations among computers also depend on the size of the problem, the number of iterations required for solution, and the precision used when printing results. In this sample problem, most values should not vary from one computer to another. Those few values that vary should generally vary only in the least significant digit; however, the budget term "IN-OUT", because it is the difference of two nearly identical numbers, may vary by as much as a factor of 3 from one computer to another.

```
                U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER MODEL

SAMPLE----3 LAYERS, 15 ROWS, 15 COLUMNS; STEADY STATE; CONSTANT HEADS COLUMN 1, LAYERS 1 AND 2; RECHARGE, WELLS AND DRAINS
   3 LAYERS        15 ROWS         15 COLUMNS
  1 STRESS PERIOD(S) IN SIMULATION
MODEL TIME UNIT IS SECONDS

I/O UNITS:
ELEMENT OF IUNIT:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
       I/O UNIT: 11 12 13  0  0  0  0 18 19  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT  1
ARRAYS RHS AND BUFF WILL SHARE MEMORY.
START HEAD WILL NOT BE SAVED -- DRAWDOWN CANNOT BE CALCULATED
    5892 ELEMENTS IN X ARRAY ARE USED BY BAS
    5892 ELEMENTS OF X ARRAY USED OUT OF   30000

BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 11
STEADY-STATE SIMULATION
     LAYER  AQUIFER TYPE
     --------------------
        1         1
        2         0
        3         0
     453 ELEMENTS IN X ARRAY ARE USED BY BCF
    6345 ELEMENTS OF X ARRAY USED OUT OF   30000

WEL1 -- WELL PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM 12
MAXIMUM OF   15 WELLS
      60 ELEMENTS IN X ARRAY ARE USED FOR WELLS
    6405 ELEMENTS OF X ARRAY USED OUT OF   30000

DRN1 -- DRAIN PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 13
MAXIMUM OF    9 DRAINS
      45 ELEMENTS IN X ARRAY ARE USED FOR DRAINS
    6450 ELEMENTS OF X ARRAY USED OUT OF   30000

RCH1 -- RECHARGE PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 18
OPTION 1 -- RECHARGE TO TOP LAYER
     225 ELEMENTS OF X ARRAY USED FOR RECHARGE
    6675 ELEMENTS OF X ARRAY USED OUT OF   30000

SIP1 -- STRONGLY IMPLICIT PROCEDURE SOLUTION PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 19
MAXIMUM OF  50 ITERATIONS ALLOWED FOR CLOSURE
 5 ITERATION PARAMETERS
    2905 ELEMENTS IN X ARRAY ARE USED BY SIP
    9580 ELEMENTS OF X ARRAY USED OUT OF   30000
```

SAMPLE----3 LAYERS, 15 ROWS, 15 COLUMNS; STEADY STATE; CONSTANT HEADS COLUMN 1, LAYERS 1 AND 2; RECHARGE, WELLS AND DRAINS

BOUNDARY ARRAY FOR LAYER 1 WILL BE READ ON UNIT 1 USING FORMAT: (15I3)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BOUNDARY ARRAY FOR LAYER 2 WILL BE READ ON UNIT 1 USING FORMAT: (15I3)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

```
                                                  BOUNDARY ARRAY =              1 FOR LAYER  3

AQUIFER HEAD WILL BE SET TO  999.99     AT ALL NO-FLOW NODES (IBOUND=0).

                                                    INITIAL HEAD =  0.0000000E+00 FOR LAYER  1

                                                    INITIAL HEAD =  0.0000000E+00 FOR LAYER  2

                                                    INITIAL HEAD =  0.0000000E+00 FOR LAYER  3

DEFAULT OUTPUT CONTROL -- THE FOLLOWING OUTPUT COMES AT THE END OF EACH STRESS PERIOD:
TOTAL VOLUMETRIC BUDGET
         HEAD

                                          COLUMN TO ROW ANISOTROPY =   1.000000

                                                          DELR =   5000.000

                                                          DELC =   5000.000

                                             HYD. COND. ALONG ROWS =  0.9999999E-03 FOR LAYER  1

                                                        BOTTOM =  -150.0000     FOR LAYER  1

                                         VERT HYD COND /THICKNESS =  0.2000000E-07 FOR LAYER  1

                                             TRANSMIS. ALONG ROWS =  0.1000000E-01 FOR LAYER  2

                                         VERT HYD COND /THICKNESS =  0.1000000E-07 FOR LAYER  2

                                             TRANSMIS. ALONG ROWS =  0.2000000E-01 FOR LAYER  3


                                            SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE
                                            -------------------------------------------

                                       MAXIMUM ITERATIONS ALLOWED FOR CLOSURE =       50
                                                  ACCELERATION PARAMETER =      1.0000
                                            HEAD CHANGE CRITERION FOR CLOSURE =    0.10000E-02
                                            SIP HEAD CHANGE PRINTOUT INTERVAL =        1

  5 ITERATION PARAMETERS CALCULATED FROM SPECIFIED WSEED = 0.00100000 :

        0.0000000E+00  0.8221720E+00  0.9683772E+00  0.9943766E+00  0.9990000E+00
```

STRESS PERIOD NO. 1, LENGTH = 86400.00

NUMBER OF TIME STEPS = 1

MULTIPLIER FOR DELT = 1.000

INITIAL TIME STEP SIZE = 86400.00

15 WELLS

LAYER	ROW	COL	STRESS RATE	WELL NO.
3	5	11	-5.0000	1
2	4	6	-5.0000	2
2	6	12	-5.0000	3
1	9	8	-5.0000	4
1	9	10	-5.0000	5
1	9	12	-5.0000	6
1	9	14	-5.0000	7
1	11	8	-5.0000	8
1	11	10	-5.0000	9
1	11	12	-5.0000	10
1	11	14	-5.0000	11
1	13	8	-5.0000	12
1	13	10	-5.0000	13
1	13	12	-5.0000	14
1	13	14	-5.0000	15

9 DRAINS

LAYER	ROW	COL	ELEVATION	CONDUCTANCE	DRAIN NO.
1	8	2	0.0000E+00	1.000	1
1	8	3	0.0000E+00	1.000	2
1	8	4	10.00	1.000	3
1	8	5	20.00	1.000	4
1	8	6	30.00	1.000	5
1	8	7	50.00	1.000	6
1	8	8	70.00	1.000	7
1	8	9	90.00	1.000	8
1	8	10	100.0	1.000	9

RECHARGE = 0.3000000E-07

31 ITERATIONS FOR TIME STEP 1 IN STRESS PERIOD 1

MAXIMUM HEAD CHANGE FOR EACH ITERATION:

HEAD CHANGE	LAYER,ROW,COL	HEAD CHANGE	LAYER,ROW,COL	HEAD CHANGE	LAYER,ROW,COL	HEAD CHANGE	LAYER,ROW,COL	HEAD CHANGE	LAYER,ROW,COL
-22.41	(3, 5, 11)	12.48	(1, 1, 15)	13.39	(3, 1, 14)	48.21	(1, 1, 15)	35.90	(3, 1, 13)
2.482	(1, 9, 14)	1.430	(3, 10, 13)	6.214	(1, 12, 14)	7.411	(3, 11, 14)	13.66	(1, 15, 15)
0.5503	(3, 8, 7)	0.4821	(2, 6, 9)	0.4711	(3, 5, 10)	2.019	(1, 11, 14)	2.302	(3, 5, 13)
0.1108	(1, 13, 12)	0.7059E-01	(3, 12, 11)	0.2819	(1, 14, 14)	0.3141	(3, 13, 14)	0.3321	(1, 15, 15)
0.7855E-02	(1, 13, 12)	0.1586E-01	(2, 11, 11)	0.1777E-01	(3, 11, 10)	0.7911E-01	(1, 14, 14)	0.8500E-01	(3, 7, 14)
0.4169E-02	(1, 13, 14)	0.2555E-02	(3, 14, 15)	0.9771E-02	(1, 14, 14)	0.1082E-01	(3, 13, 14)	0.1030E-01	(1, 15, 15)
0.2426E-03	(1, 13, 12)								

HEAD IN LAYER 1 AT END OF TIME STEP 1 IN STRESS PERIOD 1

	1 11	2 12	3 13	4 14	5 15	6	7	8	9	10
1	0.0000E+00 117.4	24.94 121.3	44.01 124.3	59.26 126.4	71.82 127.4	82.52	91.91	100.0	106.9	112.6
2	0.0000E+00 115.7	24.45 119.6	43.10 122.7	57.98 124.9	70.17 126.1	80.57	90.12	98.40	105.3	111.0
3	0.0000E+00 112.0	23.45 116.1	41.30 119.6	55.43 122.1	66.78 123.4	76.21	86.51	95.20	102.2	107.6
4	0.0000E+00 106.1	21.92 110.7	38.61 114.9	51.75 117.9	61.79 119.4	68.03	81.34	90.75	97.64	102.5
5	0.0000E+00 97.29	19.73 103.1	34.92 108.8	47.32 112.5	57.69 114.3	66.74	77.09	85.76	92.22	96.15
6	0.0000E+00 93.03	16.51 94.23	29.50 102.1	40.90 106.4	51.30 108.4	61.21	71.19	79.85	86.47	90.82
7	0.0000E+00 88.60	11.55 91.66	21.10 96.43	31.21 99.82	41.40 101.8	51.84	63.08	72.68	79.95	84.92
8	0.0000E+00 81.99	3.483 85.00	6.832 89.27	16.25 91.72	26.30 94.33	36.97	52.59	64.31	72.52	77.25
9	0.0000E+00 73.93	10.54 73.79	19.11 80.84	28.12 80.17	36.92 86.49	45.27	52.95	55.38	65.15	66.07
10	0.0000E+00 70.39	14.62 72.44	25.86 76.72	35.38 78.26	43.49 81.79	50.11	54.93	57.55	62.95	65.55
11	0.0000E+00 66.43	17.11 65.45	29.96 72.22	40.01 71.04	47.78 77.62	53.24	55.81	53.33	60.27	59.29
12	0.0000E+00 67.12	18.68 68.50	32.56 72.29	43.07 73.46	50.81 76.85	55.92	58.33	58.47	61.93	63.18
13	0.0000E+00 67.22	19.67 65.75	34.24 71.90	45.14 70.35	53.01 76.48	58.04	59.91	56.75	62.59	60.91
14	0.0000E+00 71.64	20.27 73.18	35.27 75.84	46.48 77.03	54.61 79.09	60.08	63.17	64.52	67.25	68.79
15	0.0000E+00 74.29	20.56 76.22	35.78 78.22	47.16 79.66	55.48 80.82	61.26	65.02	67.52	69.94	72.01

HEAD IN LAYER 2 AT END OF TIME STEP 1 IN STRESS PERIOD 1

	1 11	2 12	3 13	4 14	5 15	6	7	8	9	10
1	0.0000E+00 117.2	24.66 121.1	43.73 124.1	59.02 126.2	71.61 127.3	82.32	91.72	99.86	106.7	112.5
2	0.0000E+00 115.5	24.17 119.4	42.83 122.6	57.74 124.8	69.95 125.9	80.36	89.93	98.22	105.1	110.8
3	0.0000E+00 111.8	23.17 116.0	41.03 119.5	55.19 121.9	66.53 123.2	75.77	86.29	95.02	102.0	107.4
4	0.0000E+00 105.4	21.65 110.4	38.34 114.8	51.50 117.7	61.35 119.2	60.17	80.90	90.55	97.45	102.3
5	0.0000E+00 91.09	19.48 102.1	34.65 108.6	47.07 112.4	57.44 114.2	66.30	76.85	85.57	92.00	95.41
6	0.0000E+00 92.06	16.27 86.23	29.24 101.7	40.65 106.2	51.07 108.3	60.98	70.98	79.65	86.28	90.54
7	0.0000E+00 88.35	11.38 91.24	20.95 96.22	31.05 99.65	41.25 101.6	51.70	62.90	72.48	79.76	84.73
8	0.0000E+00 81.81	4.209 84.86	8.330 89.10	17.58 91.59	27.58 94.17	38.25	52.94	64.19	72.34	77.12
9	0.0000E+00 73.87	10.38 74.48	18.96 80.77	27.98 80.84	36.79 86.38	45.16	52.86	56.13	65.08	66.79
10	0.0000E+00 70.24	14.40 72.37	25.61 76.57	35.15 78.20	43.27 81.64	49.91	54.76	57.48	62.79	65.49
11	0.0000E+00 66.37	16.87 66.18	29.70 72.16	39.78 71.75	47.56 77.51	53.05	55.68	54.09	60.20	60.04
12	0.0000E+00 66.98	18.43 68.44	32.31 72.15	42.85 73.40	50.60 76.69	55.73	58.16	58.41	61.78	63.12
13	0.0000E+00 67.16	19.42 66.48	33.98 71.84	44.91 71.06	52.80 76.37	57.85	59.78	57.50	62.53	61.65
14	0.0000E+00 71.48	20.02 73.06	35.02 75.68	46.26 76.91	54.41 78.93	59.88	62.99	64.39	67.08	68.66
15	0.0000E+00 74.11	20.30 76.04	35.52 78.04	46.94 79.49	55.28 80.65	61.07	64.84	67.34	69.76	71.84

HEAD IN LAYER 3 AT END OF TIME STEP 1 IN STRESS PERIOD 1

	1 11	2 12	3 13	4 14	5 15	6	7	8	9	10
1	1.800 117.0	24.34 120.9	43.36 123.9	58.70 126.0	71.33 127.1	82.06	91.48	99.63	106.5	112.3
2	1.764 115.3	23.85 119.2	42.46 122.4	57.42 124.6	69.66 125.7	80.07	89.68	97.99	104.9	110.6
3	1.691 111.5	22.86 115.7	40.67 119.3	54.87 121.7	66.20 123.0	75.28	85.98	94.77	101.7	107.2
4	1.578 104.1	21.35 110.0	37.98 114.5	51.17 117.5	60.85 119.0	62.69	80.41	90.28	97.19	101.9
5	1.415 77.46	19.18 100.7	34.30 108.2	46.75 112.1	57.10 114.0	65.80	76.54	85.30	91.67	94.17
6	1.176 90.60	15.99 88.55	28.91 101.2	40.33 106.0	50.76 108.0	60.67	70.70	79.38	86.01	90.12
7	0.8273 87.98	11.21 90.77	20.79 95.94	30.88 99.41	41.09 101.4	51.55	62.67	72.22	79.50	84.46
8	0.4331 81.58	5.131 84.68	10.19 88.88	19.27 91.44	29.19 93.95	39.84	53.40	64.07	72.11	76.95
9	0.7543 73.81	10.22 75.31	18.82 80.72	27.84 81.64	36.66 86.24	45.06	52.78	57.03	65.02	67.64
10	1.039 70.05	14.13 72.33	25.29 76.39	34.85 78.15	42.99 81.43	49.65	54.54	57.44	62.61	65.44
11	1.224 66.33	16.59 67.06	29.37 72.13	39.47 72.60	47.28 77.38	52.79	55.53	55.01	60.16	60.94
12	1.341 66.80	18.15 68.41	31.97 71.97	42.54 73.36	50.32 76.49	55.47	57.94	58.37	61.60	63.08
13	1.415 67.12	19.14 67.35	33.65 71.80	44.61 71.90	52.53 76.24	57.60	59.63	58.39	62.48	62.54
14	1.460 71.27	19.73 72.91	34.68 75.47	45.96 76.77	54.13 78.71	59.63	62.76	64.24	66.87	68.52
15	1.481 73.87	20.01 75.82	35.18 77.81	46.63 79.27	55.00 80.42	60.81	64.59	67.11	69.52	71.61

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

CUMULATIVE VOLUMES	L**3	RATES FOR THIS TIME STEP	L**3/T
IN:		IN:	
STORAGE =	0.00000E+00	STORAGE =	0.00000E+00
CONSTANT HEAD =	0.00000E+00	CONSTANT HEAD =	0.00000E+00
WELLS =	0.00000E+00	WELLS =	0.00000E+00
DRAINS =	0.00000E+00	DRAINS =	0.00000E+00
RECHARGE =	0.13608E+08	RECHARGE =	157.50
TOTAL IN =	0.13608E+08	TOTAL IN =	157.50
OUT:		OUT:	
STORAGE =	0.00000E+00	STORAGE =	0.00000E+00
CONSTANT HEAD =	0.43265E+07	CONSTANT HEAD =	50.075
WELLS =	0.64800E+07	WELLS =	75.000
DRAINS =	0.28011E+07	DRAINS =	32.420
RECHARGE =	0.00000E+00	RECHARGE =	0.00000E+00
TOTAL OUT =	0.13608E+08	TOTAL OUT =	157.49
IN - OUT =	184.00	IN - OUT =	0.21210E-02
PERCENT DISCREPANCY =	0.00	PERCENT DISCREPANCY =	0.00

TIME SUMMARY AT END OF TIME STEP 1 IN STRESS PERIOD 1

	SECONDS	MINUTES	HOURS	DAYS	YEARS
TIME STEP LENGTH	86400.0	1440.00	24.0000	1.00000	0.273785E-02
STRESS PERIOD TIME	86400.0	1440.00	24.0000	1.00000	0.273785E-02
TOTAL SIMULATION TIME	86400.0	1440.00	24.0000	1.00000	0.273785E-02

APPENDIX E

ABBREVIATED INPUT INSTRUCTIONS

These input instructions are intended as a quick reference for the experienced user. Most explanations that are contained in the complete input instructions given in package documentation have been omitted. The format of input fields is given only for those records that contain fields that are not 10 characters wide. Each input item, for which format is not given, is identified as either a record or an array. For records, the fields contained in the record are named. For arrays, only the array name is given. Input fields which contain codes or flags are described. All other field and array descriptions have been dropped.

Array Input

The real two-dimensional array reader (U2DREL), the integer two-dimensional array reader (U2DINT), and the real one-dimensional array reader (U1DREL) read one array-control record and, optionally, a data array in a format specified on the array-control record.

FOR REAL ARRAY READER (U2DREL or U1DREL)

Data:	LOCAT	CNSTNT	FMTIN	IPRN
Format:	I10	F10.0	5A4	I10

FOR INTEGER ARRAY READER (U2DINT)

Data:	LOCAT	ICONST	FMTIN	IPRN
Format:	I10	I10	5A4	I10

IPRN--is a flag indicating that the array being read should be printed and a code for indicating the format that should be used. It is used only if LOCAT is not equal to zero. The format codes are different for each of the three modules. IPRN is set to zero when the specified value exceeds those defined in the chart below. If IPRN is less than zero, the array will not be printed.

IPRN	U2DREL	U2DINT	U1DREL
0	10G11.4	10I11	10G12.5
1	11G10.3	60I1	
2	9G13.6	40I2	
3	15F7.1	30I3	
4	15F7.2	25I4	
5	15F7.3	20I5	
6	15F7.4		
7	20F5.0		
8	20F5.1		
9	20F5.2		
10	20F5.3		
11	20F5.4		
12	10G11.4		

LOCAT--indicates the location of the data which will be put in the array.
If LOCAT < 0, unit number for unformatted records.
If LOCAT = 0, all elements are set equal to CNSTNT or ICONST.
If LOCAT > 0, unit number for formatted records.

Basic Package Input

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

```
FOR EACH SIMULATION
            1.  Record: HEADNG(32)
            2.  Record: HEADNG (continued)
            3.  Record: NLAY    NROW    NCOL    NPER    ITMUNI
            4.  Data:   IUNIT(24)
                Format: 24I3
                   (BCF WEL DRN RIV EVT XXX GHB RCH SIP XXX SOR OC)
                     1   2   3   4   5   6   7   8   9   10  11 12
            5.  Record: IAPART  ISTRT
            6.  Array:  IBOUND(NCOL,NROW)
                        (One array for each layer in the grid)
            7.  Record: HNOFLO
            8.  Array:  Shead(NCOL,NROW)
                        (One array for each layer in the grid)
FOR EACH STRESS PERIOD
            9.  Data:   PERLEN   NSTP  TSMULT
```

ITMUNI--is the time unit of model data.

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

Consistent length and time units must be used for all model data. The user may choose one length unit and one time unit to be used to specify all input data.

IUNIT--is a 24-element table of input units for use by all major options.

IAPART--indicates whether array BUFF is separate from array RHS.
If IAPART = 0, the arrays BUFF and RHS occupy the same space. This option conserves space. This option should be used unless some other package explicitly says otherwise.
If IAPART ≠ 0, the arrays BUFF and RHS occupy different space.

ISTRT--indicates whether starting heads are to be saved.
If ISTRT = 0, starting heads are not saved.
If ISTRT ≠ 0, starting heads are saved.

IBOUND--is the boundary array.
If IBOUND(I,J,K) < 0, cell I,J,K has a constant head.
If IBOUND(I,J,K) = 0, cell I,J,K is inactive.
If IBOUND(I,J,K) > 0, cell I,J,K is active.

HNOFLO--is the value of head to be assigned to all inactive cells.

Shead--is head at the start of the simulation.

PERLEN--is the length of a stress period.

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

TSMULT--is the multiplier for the length of successive time steps.

Output Control Input

Input to Output Control is read from the unit specified in IUNIT(12). All printer output goes to unit 6 as specified in the main program. If necessary, the unit number for printer output can be changed to meet the requirements of a particular computer.

```
FOR EACH SIMULATION
              1. Record:  IHEDFM     IDDNFM      IHEDUN     IDDNUN
FOR EACH TIME STEP
              2. Record:  INCODE  IHDDFL  IBUDFL  ICBCFL
              3. Record:  Hdpr    Ddpr    Hdsv    Ddsv
(Record 3 is read 0, 1, or NLAY times, depending on the value of INCODE.)
```

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

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

IHEDUN--is the unit number on which heads will be saved.
IDDNUN--is the unit number on which drawdowns will be saved.
INCODE--is the head/drawdown ouput code.
If INCODE < 0, layer-by-layer specifications from the last time steps are used. Input item 3 is not read.
If INCODE = 0, all layers are treated the same way. Input item 3 will consist of one record. IOFLG array will be read.
If INCODE > 0, input item 3 will consist of one record for each layer.
IHDDFL--is a head and drawdown output flag.
If IHDDFL = 0, neither heads nor drawdowns will be printed or saved.
If IHDDFL ≠ 0, heads and drawdowns will be printed or saved.
IBUDFL--is a budget print flag.
If IBUDFL = 0, overall volumetric budget will not be printed.
If IBUDFL ≠ 0, overall volumetric budget will be printed.
ICBCFL--is a cell-by-cell flow-term flag.
If ICBCFL = 0, cell-by-cell flow terms are not saved or printed.
If ICBCFL ≠ 0, cell-by-cell flow terms are printed or recorded on disk depending on flags set in the component of flow packages, i.e., IWELCB, IRCHCB, etc.
Hdpr--is the output flag for head printout.
If Hdpr = 0, head is not printed for the corresponding layer.
If Hdpr ≠ 0, head is printed for the corresponding layer.
Ddpr--is the output flag for drawdown printout.
If Ddpr = 0, drawdown is not printed for the corresponding layer.
If Ddpr ≠ 0, drawdown is printed for the corresponding layer.
Hdsv--is the output flag for head save.
If Hdsv = 0, head is not saved for the corresponding layer.
If Hdsv ≠ 0, head is saved for the corresponding layer.
Ddsv--is the output flag for drawdown save.
If Ddsv = 0, drawdown is not saved for the corresponding layer.
If Ddsv ≠ 0, drawdown is saved for the corresponding layer.

Block-Centered Flow Package Input

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

FOR EACH SIMULATION

1. Record: ISS IBCFCB
2. Data: LAYCON(NLAY) (maximum of 80 layers)
   Format: 40I2
   (If there are 40 or fewer layers, use one record.)

3. Array: TRPY(NLAY)
4. Array: DELR(NCOL)
5. Array: DELC(NROW)

All of the arrays (items 6-12) for layer 1 are read first; then all of the arrays for layer 2, etc.

IF THE SIMULATION IS TRANSIENT

6. Array: sf1(NCOL,NROW)

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

7. Array: Tran(NCOL,NROW)

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

8. Array: HY(NCOL,NROW)
9. Array: BOT(NCOL,NROW)

IF THIS IS NOT THE BOTTOM LAYER

10. Array: Vcont(NCOL,NROW)

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

11. Array: sf2(NCOL,NROW)

IF THE LAYER TYPE CODE IS TWO OR THREE

12. Array: TOP(NCOL,NROW)

ISS--is the steady-state flag.
If ISS ≠ 0, the simulation is steady state.
If ISS = 0, the simulation is transient.

IBCFCB--is a flag and a unit number.
If IBCFCB > 0, cell-by-cell flow terms will be recorded if ICBCFL (see Output Control) is set.
If IBCFCB = 0, cell-by-cell flow terms will not be printed or recorded.
If IBCFCB < 0, print flow for constant-head cells if ICBCFL is set.

LAYCON--is the layer type table: 0 - confined, 1 - unconfined, 2 - confined/unconfined (T constant), and 3 - confined/unconfined.

TRPY--is an anisotropy factor for each layer: T or K along a column to T or K along a row.

DELR--is the cell width along rows.

DELC--is the cell width along columns.

sf1--is the primary storage factor.

Tran--is the transmissivity along rows.

HY--is the hydraulic conductivity along rows.

BOT--is the elevation of the aquifer bottom.

Vcont--is the vertical hydraulic conductivity divided by the thickness from a layer to the layer beneath it.

sf2--is the secondary storage factor.

TOP--is the elevation of the aquifer top.

River Package Input

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

```
FOR EACH SIMULATION
            1.  Record: MXRIVR  IRIVCB
FOR EACH STRESS PERIOD
            2.  Record: ITMP
            3.  Record: Layer   Row     Column  Stage   Cond    Rbot
                (Input item 3 normally consists of one record for each
                river reach.  If ITMP is negative or zero, item 3 is not
                read.)
```

IRIVCB--is a flag and a unit number.
If IRIVCB > 0, cell-by-cell flow terms will be recorded.
If IRIVCB = 0, cell-by-cell flow terms will not be printed or recorded.
If IRIVCB < 0, river leakage will be printed if ICBCFL is set.

ITMP--is a flag and a counter.
If ITMP < 0, river data from the last stress period will be reused.
If ITMP $\geq$ 0, ITMP will be the number of reaches active during the current stress period.

Recharge Package Input

Input to the Recharge (RCH) Package is read from the unit specified in IUNIT(8).

```
FOR EACH SIMULATION
            1.  Record: NRCHOP   IRCHCB
FOR EACH STRESS PERIOD
            2.  Record: INRECH   INIRCH
            3.  Array:  RECH(NCOL,NROW)
IF THE RECHARGE OPTION IS EQUAL TO 2
            4.  Array:  IRCH(NCOL,NROW)
```

NRCHOP--is the recharge option code.
1 - Recharge is only to the top grid layer.
2 - Vertical distribution of recharge is specified in array IRCH.
3 - Recharge is applied to the highest active cell in each vertical column.

IRCHCB--is a flag and a unit number.
If IRCHCB > 0, unit number for cell-by-cell flow terms.
If IRCHCB $\leq$ 0, cell-by-cell flow terms will not be printed or recorded.

INRECH--is the RECH read flag.
If INRECH < 0, recharge fluxes from the preceding stress period are used.
If INRECH $\geq$ 0, an array of recharge fluxes, RECH (Lt^{-1}), is read.

INIRCH--is similar to INRECH.

Well Package Input

Input for the Well (WEL) Package is read from the unit specified in IUNIT(2).

```
FOR EACH SIMULATION
            1. Record: MXWELL  IWELCB
FOR EACH STRESS PERIOD
            2. Record: ITMP
            3. Record: Layer       Row       Column       Q
               (Input item 3 normally consists of one record for each
               well.  If ITMP is negative or zero, item 3 is not read.)
```

MXWELL--is the maximum number of wells used at any time.
IWELCB--is a flag and a unit number.
If IWELCB > 0, unit number for cell-by-cell flow terms.
If IWELCB = 0, cell-by-cell flow terms will not be printed or recorded.
If IWELCB < 0, well recharge will be printed whenever ICBCFL is set.
ITMP--is a flag and a counter.
If ITMP < 0, well data from the last stress period will be reused.
If ITMP $\geq$ 0, ITMP will be the number of wells active during the current stress period.

Drain Package Input

Input to the Drain (DRN) Package is read from the unit specified in IUNIT(3).

```
FOR EACH SIMULATION
            1.  Record:  MXDRN      IDRNCB
FOR EACH STRESS PERIOD
            2.  Record:  ITMP
            3.  Record:  Layer     Row     Col     Elevation     Cond
            (Input item 3 normally consists of one record for each drain.
            If ITMP is negative or zero, item 3 will not be read.)
```

MXDRN--is the maximum number of drain cells active at one time.
IDRNCB--is a flag and a unit number.
If IDRNCB > 0, unit number for cell-by-cell flow terms.
If IDRNCB = 0, cell-by-cell flow terms will not be printed or recorded.
If IDRNCB < 0, drain leakage for each cell will be printed whenever ICBCFL is set.
ITMP--is a flag and a counter.
If ITMP < 0, drain data from the last stress period will be reused.
If ITMP $\geq$ 0, ITMP will be the number of drains active during the current stress period.

Evapotranspiration Package Input

Input to the Evapotranspiration (EVT) Package is read from the unit specified in IUNIT (5).

FOR EACH SIMULATION
1. Record: NEVTOP IEVTCB

FOR EACH STRESS PERIOD
2. Record: INSURF INEVTR INEXDP INIEVT
3. Array: SURF
4. Array: EVTR
5. Array: EXDP

IF THE ET OPTION IS EQUAL TO TWO
6. Array: IEVT

NEVTOP--is the evapotranspiration (ET) option code.
1 - ET is calculated only for cells in the top grid layer.
2 - The cell for each vertical column is specified by the user in array IEVT.

IEVTCB--is a flag and a unit number.
If IEVTCB > 0, unit number for cell-by-cell flow terms.
If IEVTCB $\leq$ 0, cell-by-cell flow terms will not be printed or recorded.

INSURF--is the ET surface (SURF) read flag.
If INSURF $\geq$ 0, an array containing the ET surface elevation will be read.
If INSURF < 0, the ET surface from the preceding stress period will be reused.

INEVTR--is similar to INSURF.

INEXDP--is similar to INSURF.

INIEVT--is similar to INSURF.

General-Head Boundary Package Input

Input for the General-Head Boundary (GHB) Package is read from the unit specified in IUNIT(7).

```
FOR EACH SIMULATION
    1. Record:  MXBND      IGHBCB
FOR EACH STRESS PERIOD
    2. Record:  ITMP                                    Boundary
    3. Record:  Layer       Row       Column          Head          Cond
        (Input item 3 normally consists of one record for each GHB.
        If ITMP is negative or zero, item 3 is not read.)
```

MXBND--is the maximum number of general-head boundary cells at one time.

IGHBCB--is a flag and a unit number.
- If IGHBCB > 0, unit number for cell-by-cell flow terms.
- If IGHBCB = 0, cell-by-cell flow terms will not be printed or recorded.
- If IGHBCB < 0, boundary leakage for each cell will be printed whenever ICBCFL is set.

ITMP--is a flag and a counter.
- If ITMP < 0, GHB data from the preceding stress period will be reused.
- If ITMP $\geq$ 0, ITMP is the number of general-head boundaries during the current stress period.

Strongly Implicit Procedure Package Input

Input to the Strongly Implicit Procedure (SIP) Package is read from the unit specified in IUNIT(9).

```
FOR EACH SIMULATION
    1. Record:  MXITER     NPARM
    2. Record:  ACCL       HCLOSE      IPCALC      WSEED      IPRSIP
```

IPCALC--is a flag indicating where the iteration parameter seed will come from.
- 0 - the seed will be entered by the user.
- 1 - the seed will be calculated at the start of the simulation from problem parameters.

IPRSIP--is the printout interval for SIP.

Slice-Successive Overrelaxation Package Input

Input to the Slice-Successive Overrelaxation (SOR) Package is read from the unit specified in IUNIT(11).

```
FOR EACH SIMULATION
            1. Record: MXITER
            2. Record: ACCL     HCLOSE    IPRSOR
```

IPRSOR--is the printout interval for SOR.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1988—201-985/83961