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

ILLUSTRATIONS

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 Implemementation 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

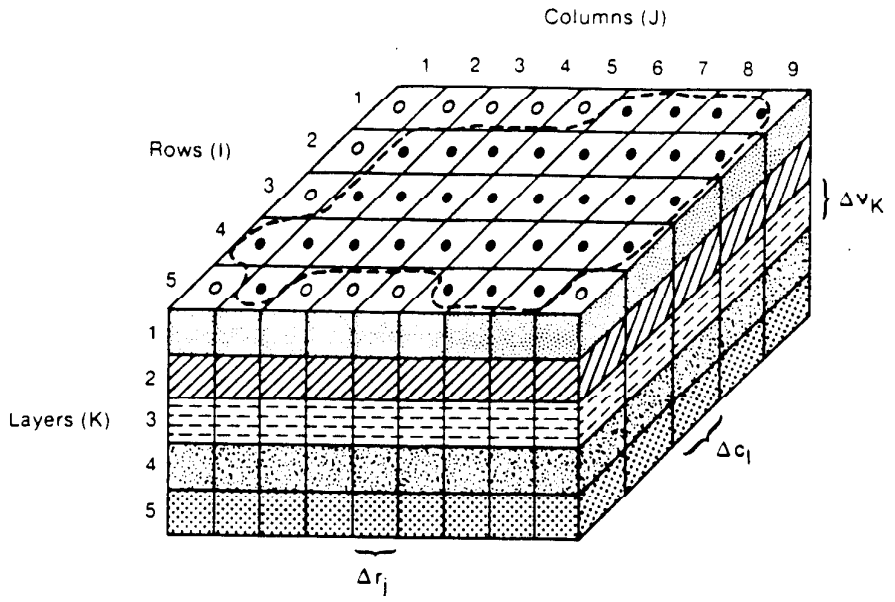
respect to space and time are substituted into equation (1), the equation and its initial and boundary conditions are satisfied. A time-varying head distribution of this nature characterizes the flow system, in that it measures both the energy of flow and the volume of water in storage, and can be used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

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

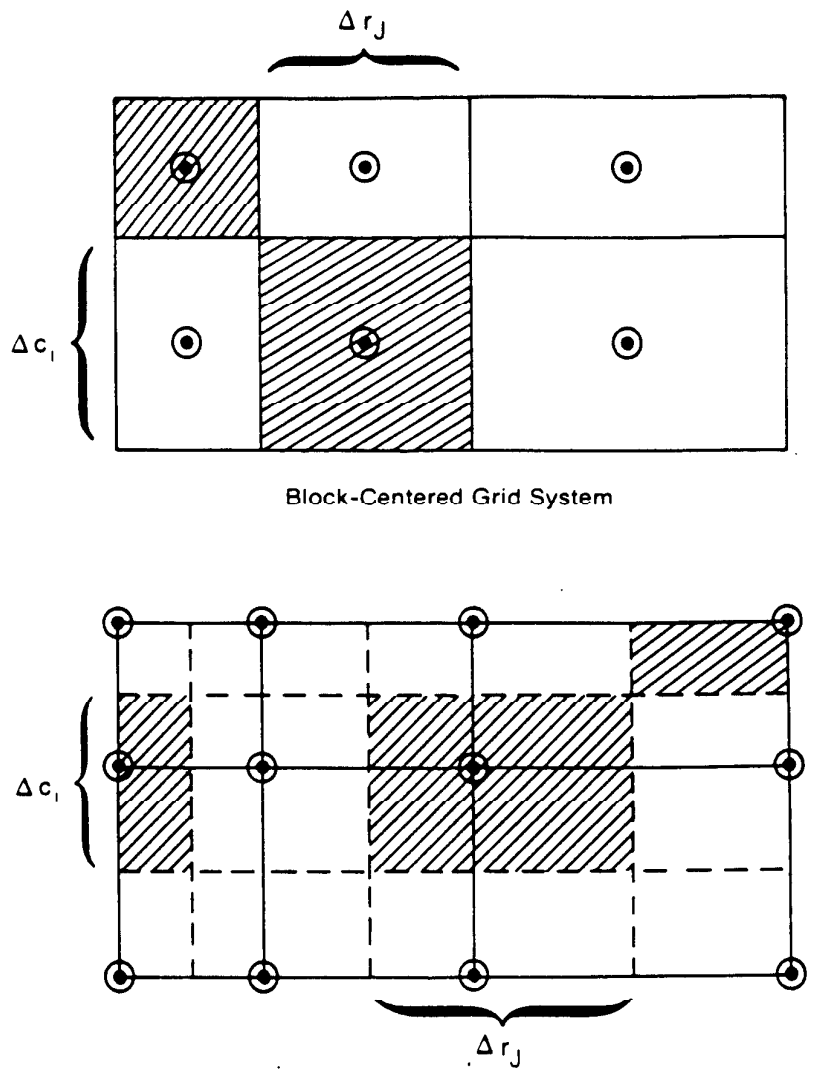


Figure 2.—Grids showing the difference between block-centered and point-centered grids.

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});

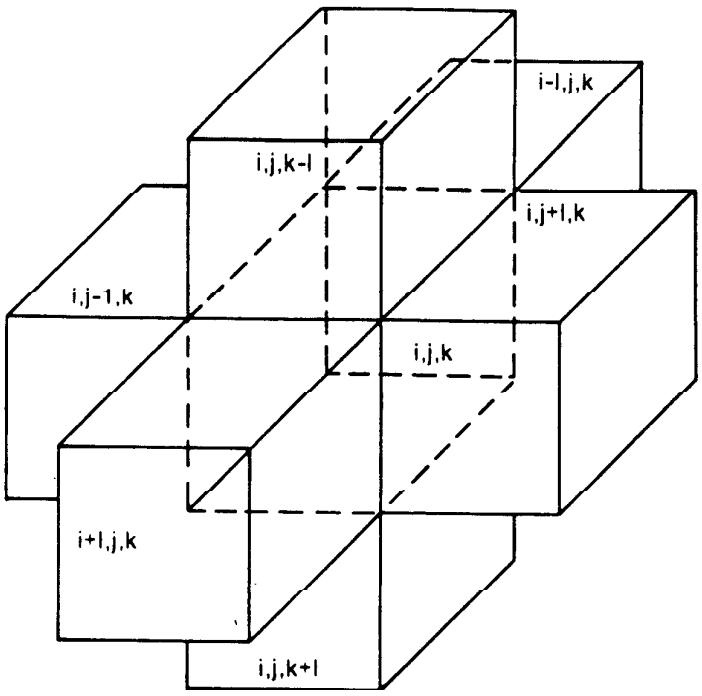


Figure 3.—Cell i,j,k and indices for the six adjacent cells.

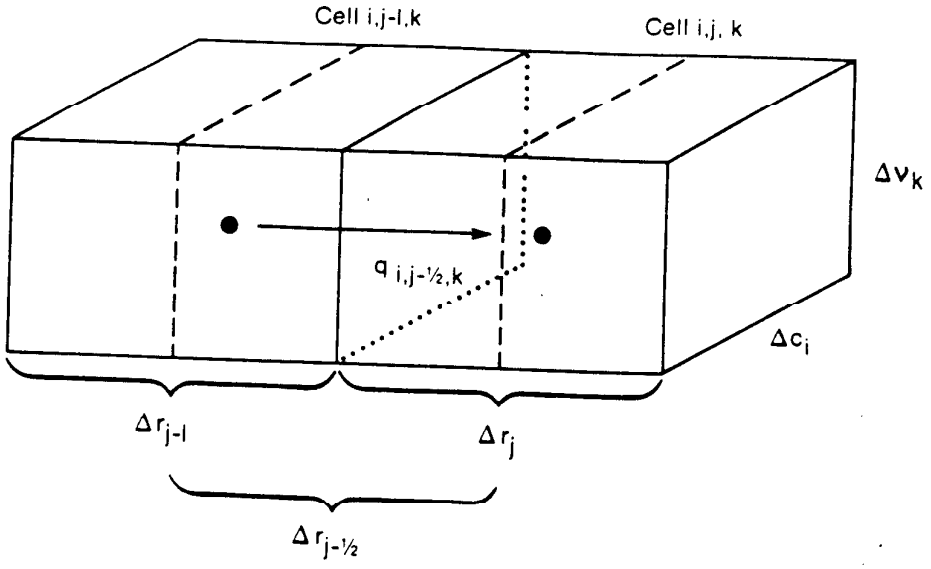


Figure 4.—Flow into cell i,j,k from cell i,j-1,k.

$KR_{i,j-1/2,k}$ 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, $-CRIV_{1,J,k,2}$ corresponds to $p_{1,j,k,2}$ 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

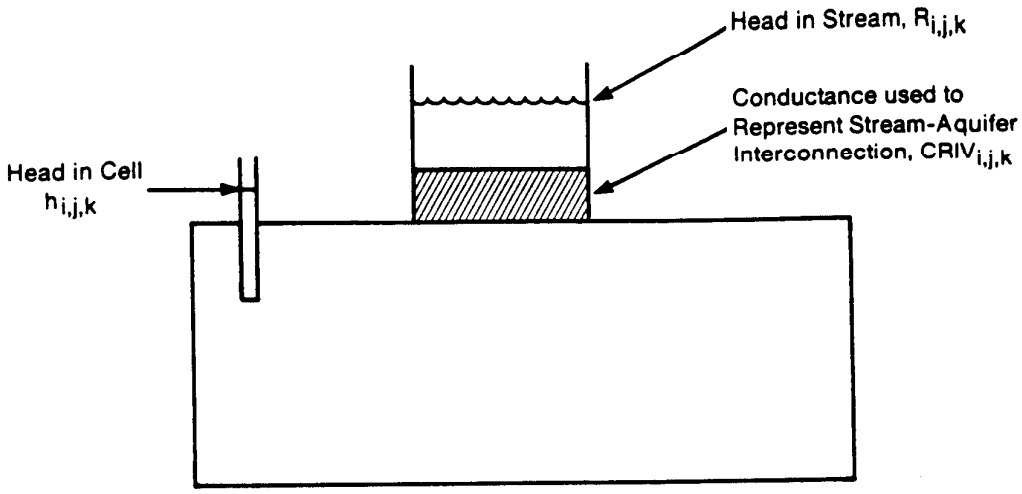


Figure 5.—Conceptual representation of leakage through a riverbed into a cell.

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

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

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

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

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

$$q_{i,j-1/2,k} + q_{i,j+1/2,k} + q_{i-1/2,j,k} + q_{i+1/2,j,k}$$

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

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).

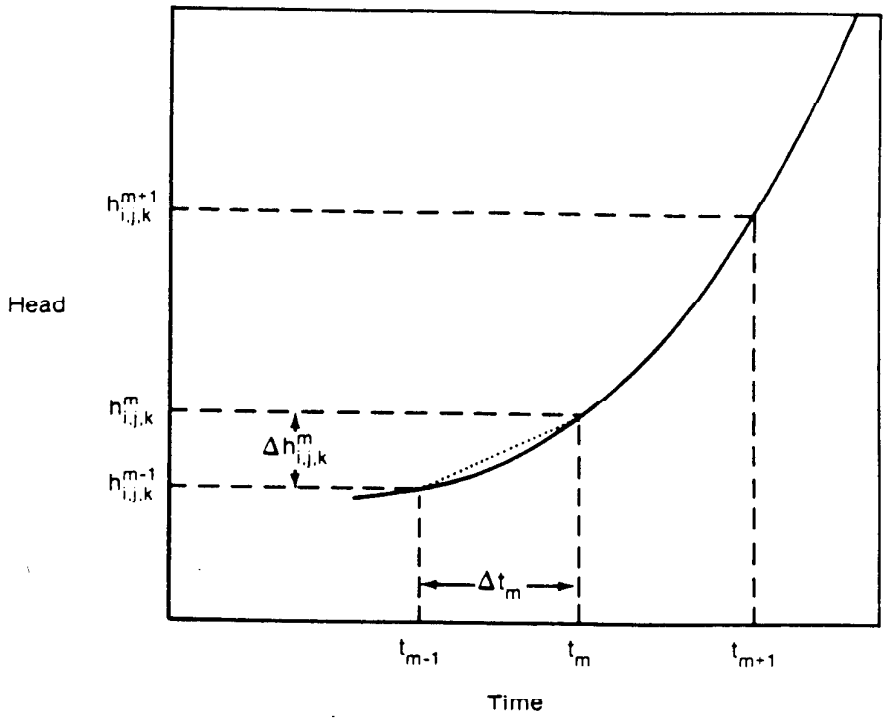
Equations (10) through (15) and (21) may be substituted into equation (22) to give the finite-difference approximation for cell i,j,k as

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

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

$$\begin{aligned}
&CR_{i,j-1/2,k}(h^2_{i,j-1,k} - h^2_{i,j,k}) + CR_{i,j+1/2,k}(h^2_{i,j+1,k} - h^2_{i,j,k}) \\
&+ CC_{i-1/2,j,k}(h^2_{i-1,j,k} - h^2_{i,j,k}) + CC_{i+1/2,j,k}(h^2_{i+1,j,k} - h^2_{i,j,k}) \\
&+ CV_{i,j,k-1/2}(h^2_{i,j,k-1} - h^2_{i,j,k}) + CV_{i,j,k+1/2}(h^2_{i,j,k+1} - h^2_{i,j,k}) \\
&+ P_{i,j,k}h^2_{i,j,k} + Q_{i,j,k} \\
&\qquad = SS_{i,j,k}\frac{(\Delta r_j \Delta c_i \Delta v_k)(h^2_{i,j,k} - h^1_{i,j,k})}{t_2 - t_1}
\end{aligned} \tag{25}$$

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

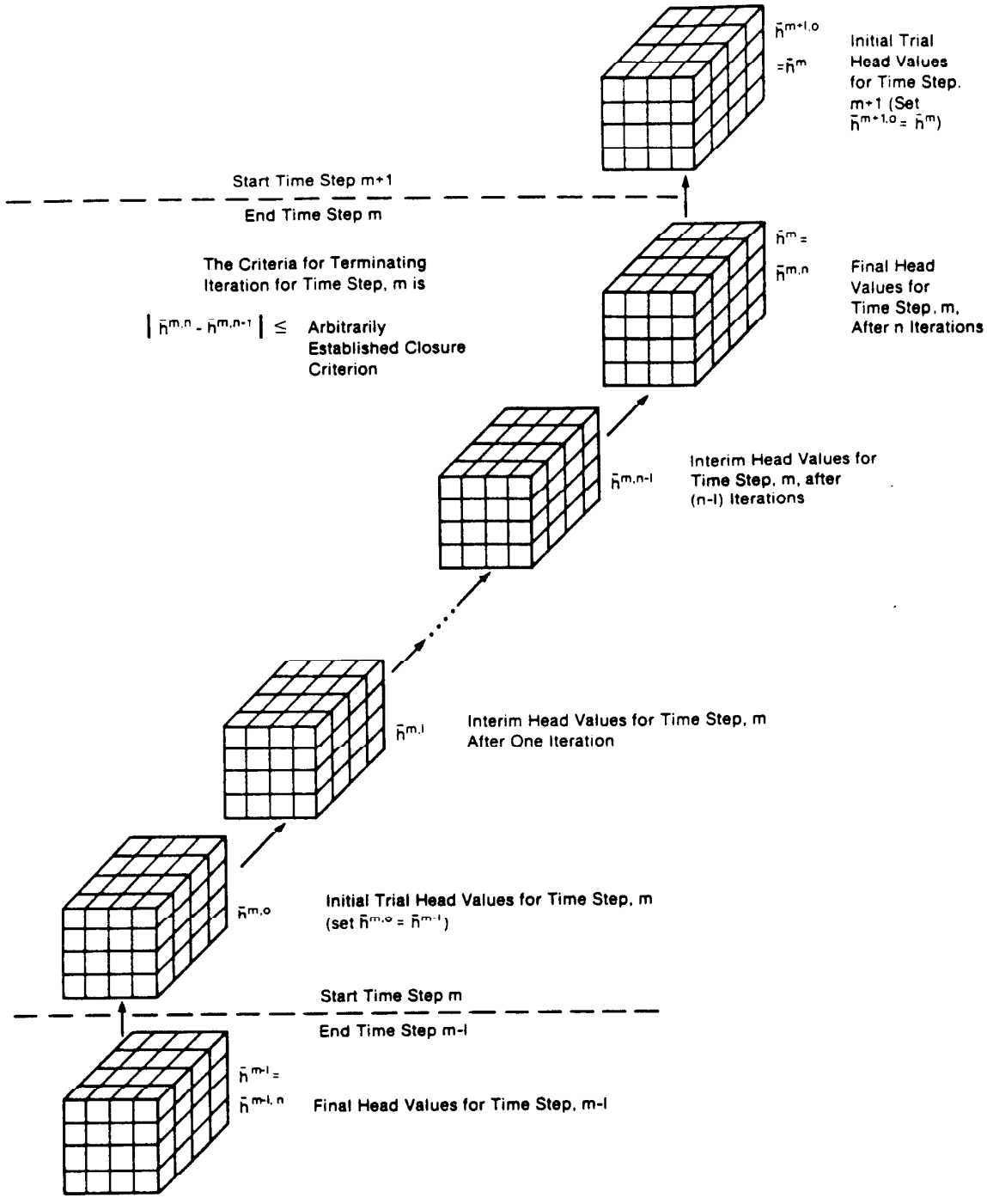


Figure 7.—Iterative calculation of a head distribution.

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.

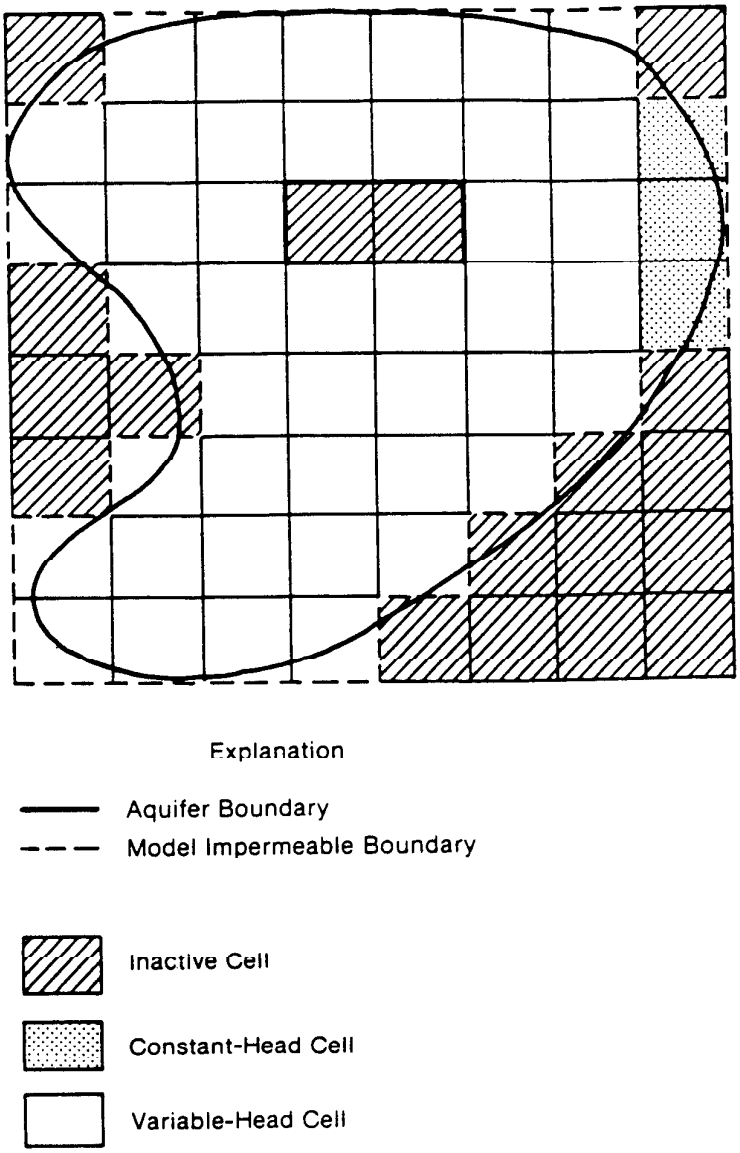
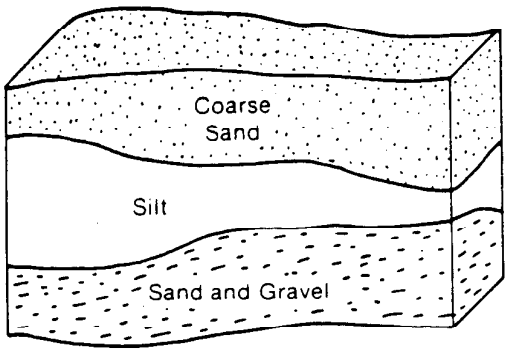


Figure 8.—Discretized aquifer showing boundaries and cell designations.

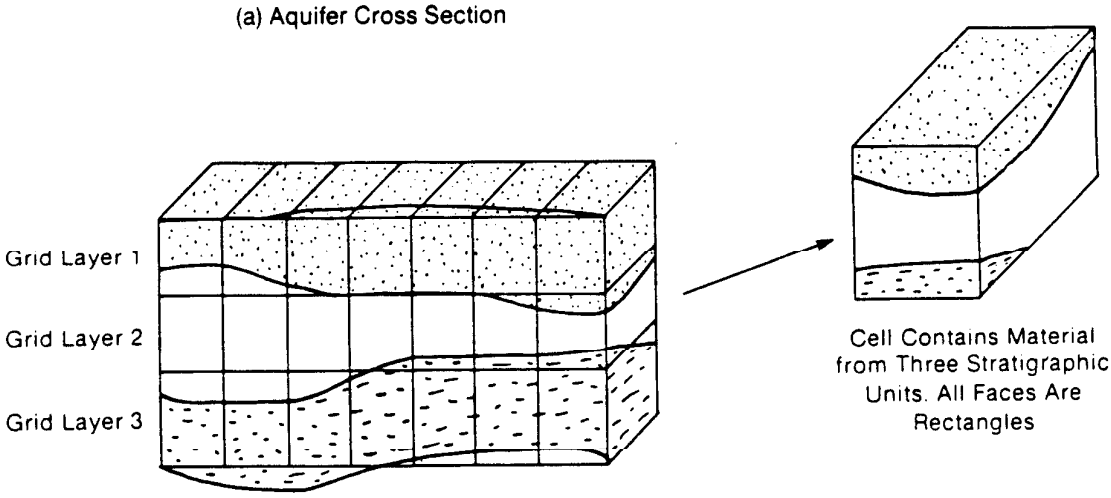
Conceptual Aspects of Vertical Discretization

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.

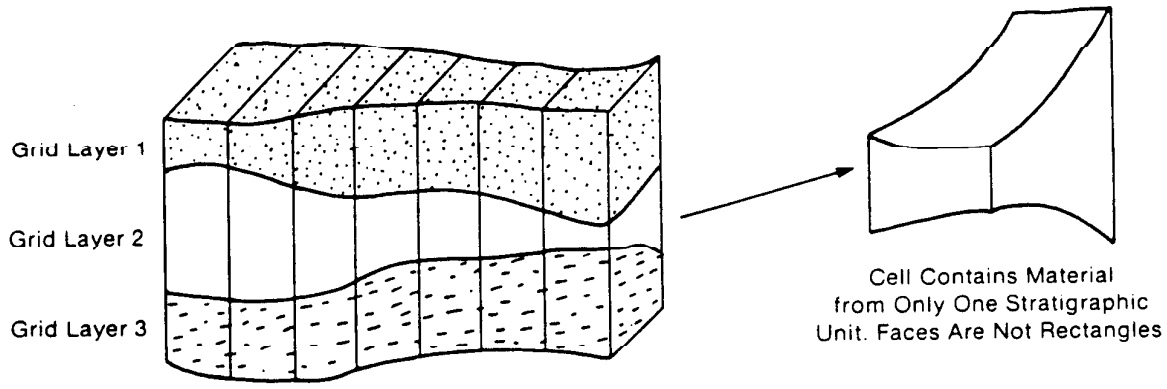
At one extreme, vertical discretization can be visualized simply as an extension of areal discretization--a more or less arbitrary process of dividing the flow system into segments along the vertical, governed in part by the vertical resolution desired in the results. At the opposite extreme, vertical 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

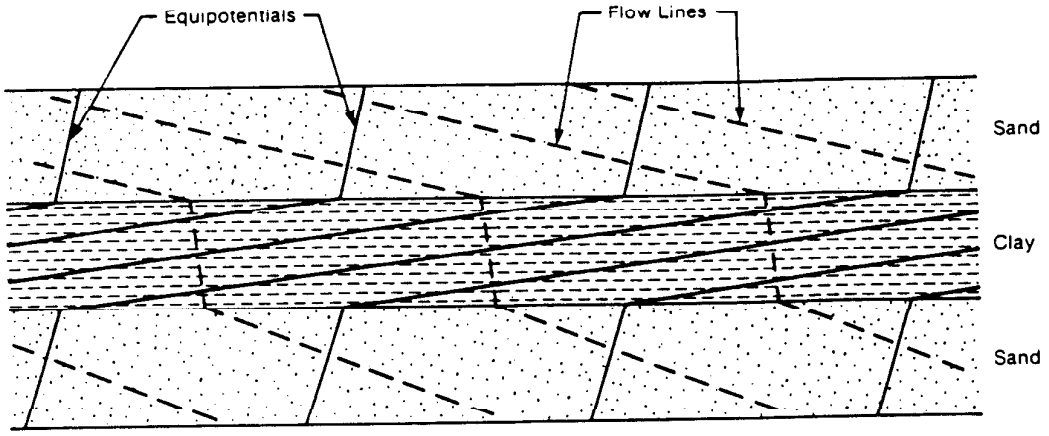


Figure 10.—Possible pattern of flow in a cross section consisting of two high conductivity units separated by a low conductivity unit.

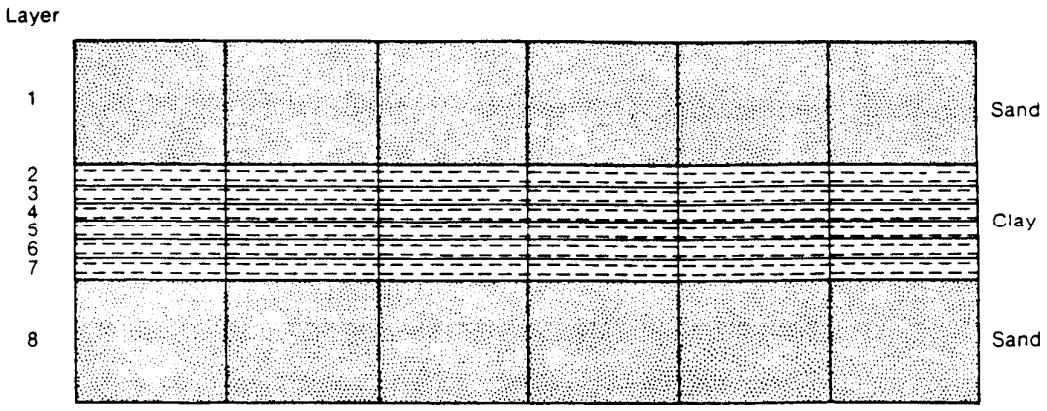


Figure 11.—A cross section in which a low conductivity unit is represented by six model layers.

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.

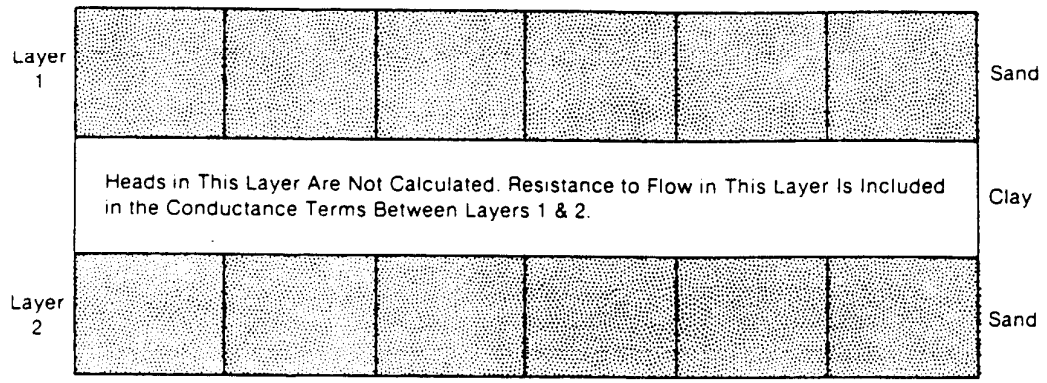


Figure 12.—A cross section in which a low conductivity unit is represented by the conductance between model layers.

CHAPTER 3

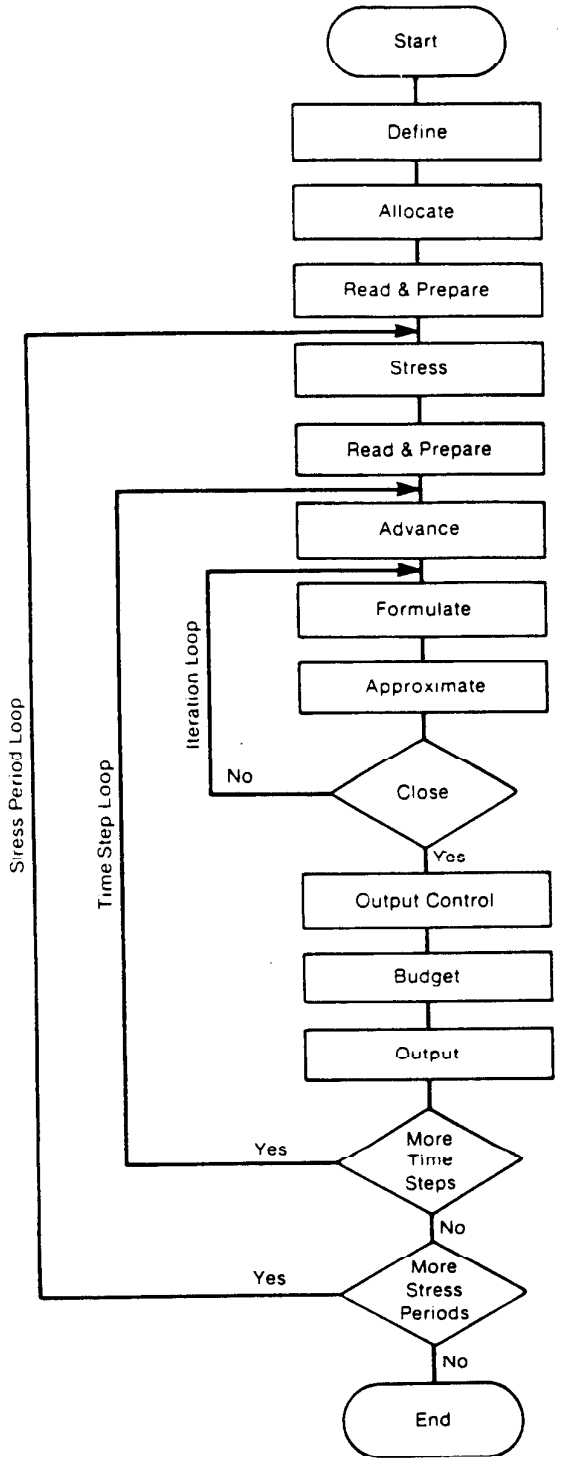
PROGRAM DESIGN

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

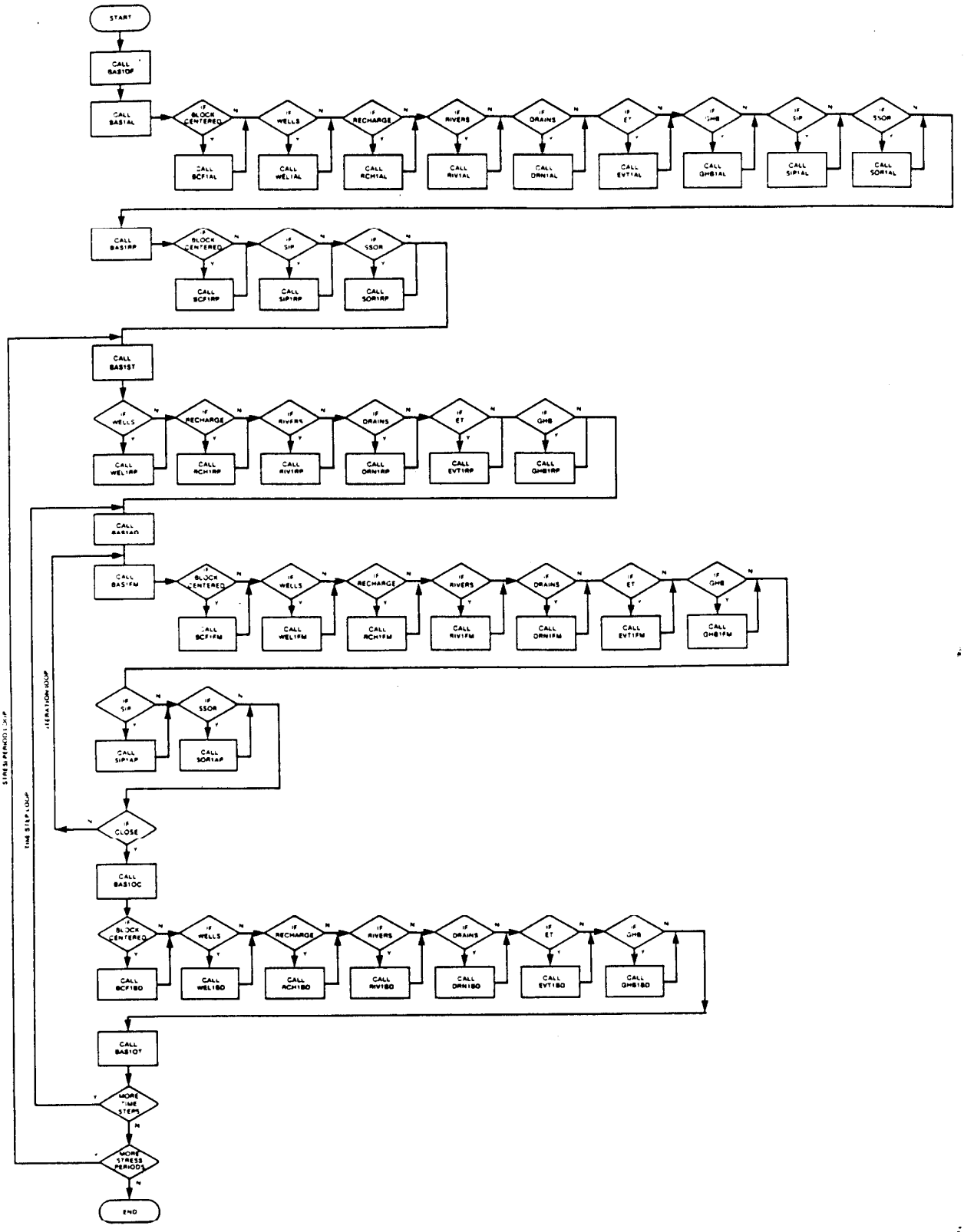


Figure 16.—Overall program structure showing all primary modules.

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 any 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.

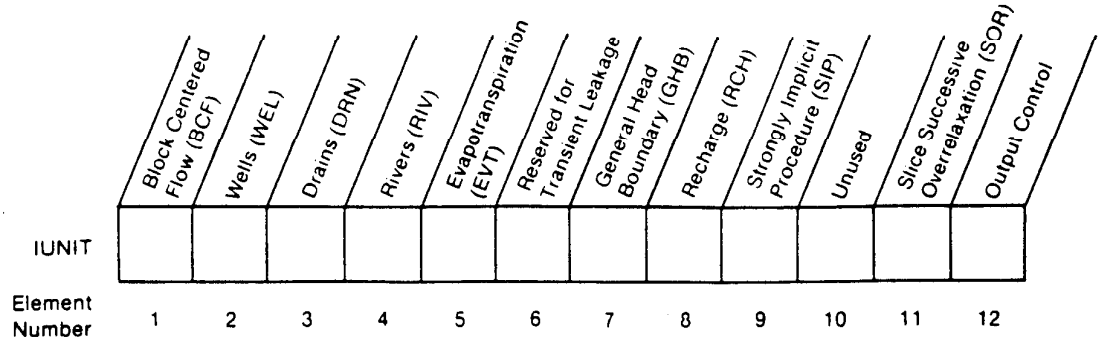
For input purposes, the program may be discussed in terms of "major options"; these are major segments of the program which are utilized only at the user's request. They correspond generally to the individual packages; in fact, all of the existing packages except the Basic Package are considered major 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

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

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

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

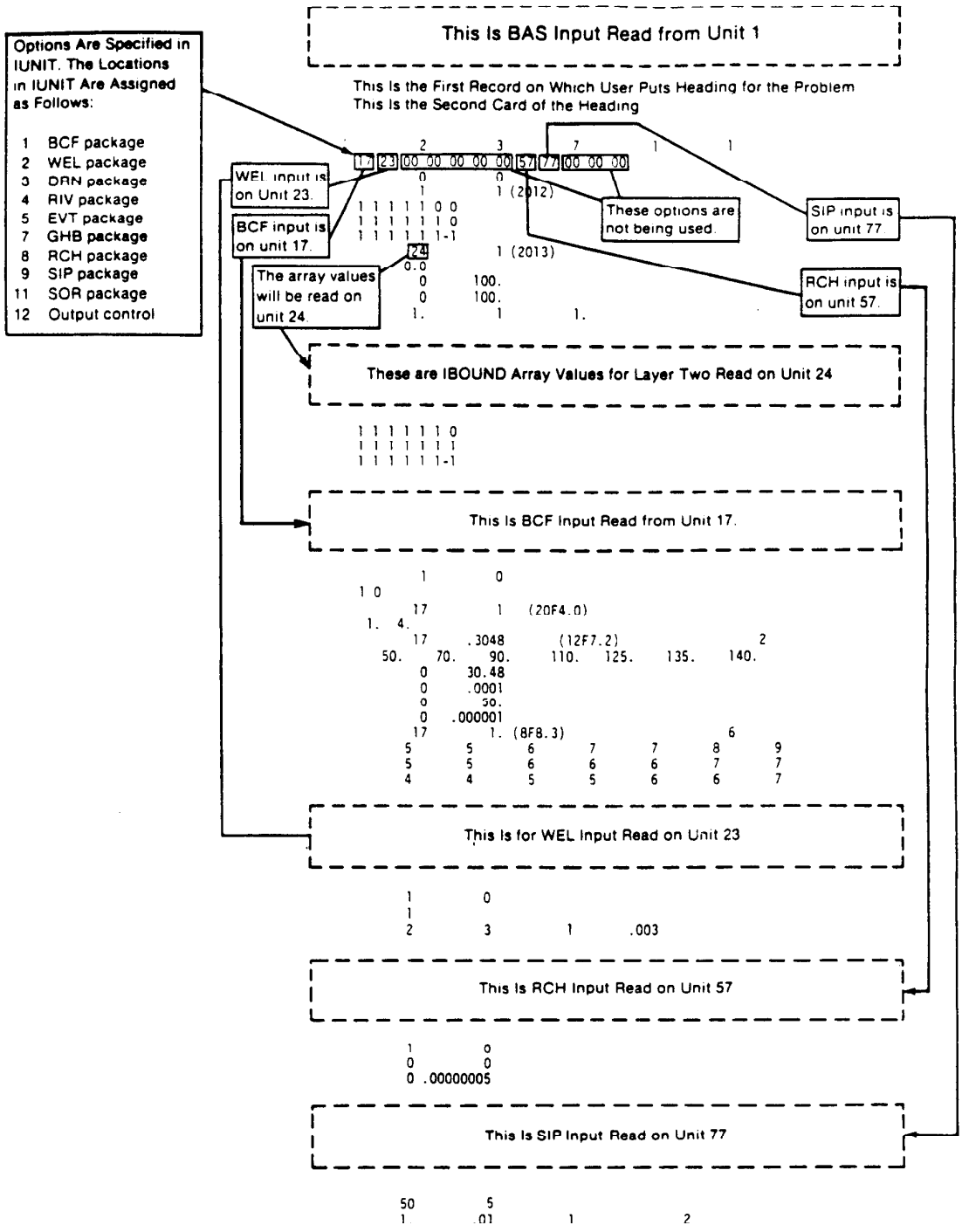


Figure 18.—Sample input data showing role of the IUNIT array.

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

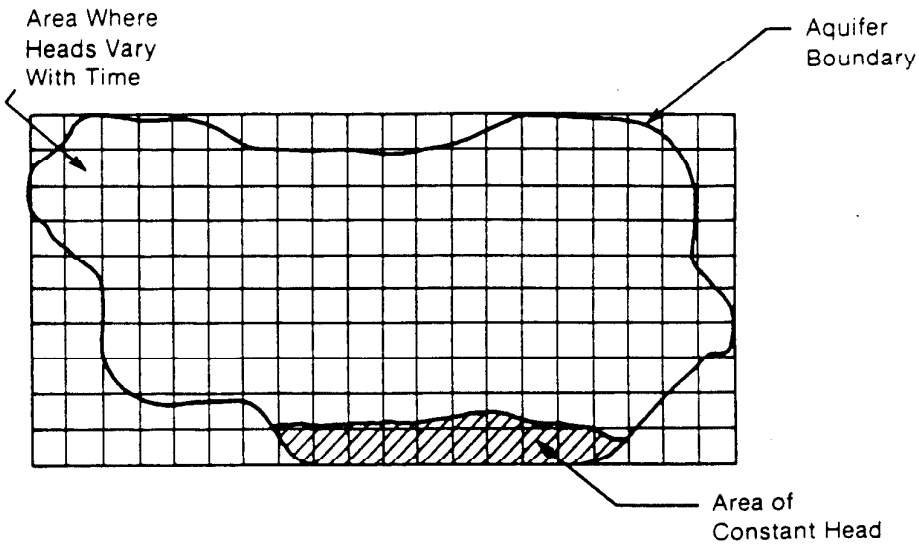
Recall that the finite-difference equation for a cell has the form

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

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.

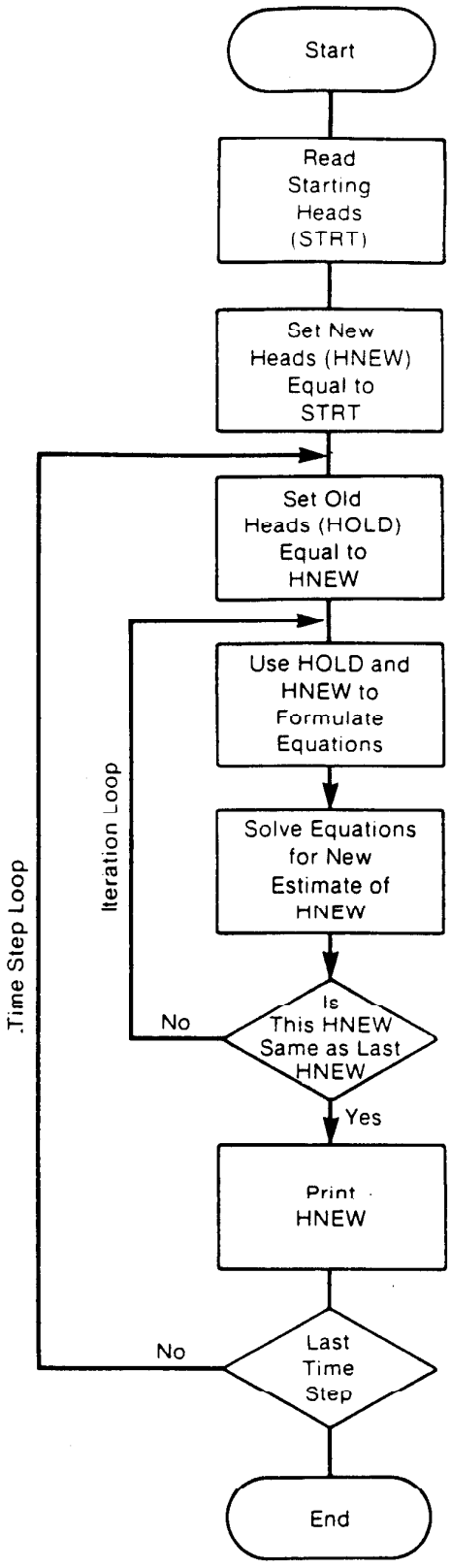


Figure 20.—Flow of head distributions during a simulation.

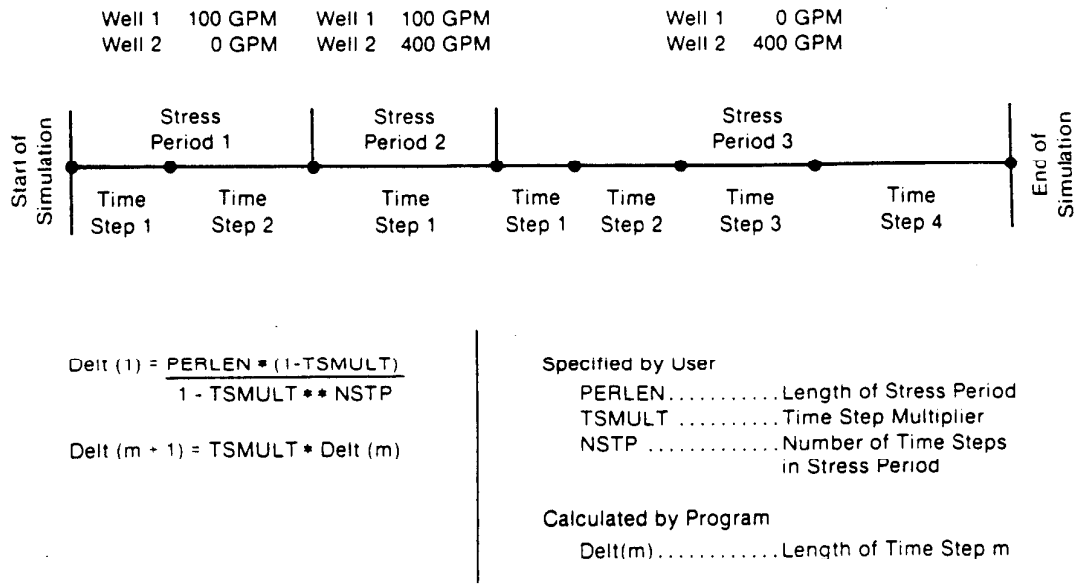
calculate drawdown, the difference between the starting head distribution and some later head distribution.

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

BAS1OC

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

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

Explanation of Fields Used in Input Instructions

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

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

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

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

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

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

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

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

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

IHDDFL--is a head and drawdown output flag.

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

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

IBUDFL--is a budget print flag.

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

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

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

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

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

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

Hdpr--is the output flag for head printout.

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

If Hdpr ≠ 0, head is printed for the corresponding layer.

Ddpr--is the output flag for drawdown printout.

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

If Ddpr ≠ 0, drawdown is printed for the corresponding layer.

Hdsv--is the output flag for head save.

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

If Hdsv ≠ 0, head is saved for the corresponding layer.

Ddsv--is the output flag for drawdown save.

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

If Ddsv ≠ 0, drawdown is saved for the corresponding layer.

SAMPLE INPUT TO THE OUTPUT CONTROL OPTION

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

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.

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

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

The CV, CR, and CC coefficients are conductances between nodes--sometimes called "branch conductances." The HCOF and RHS coefficients are composed of external source terms and storage terms. Besides calculating the conductances and storage terms, the BCF Package calculates flow-correction terms that are added to HCOF and RHS 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);

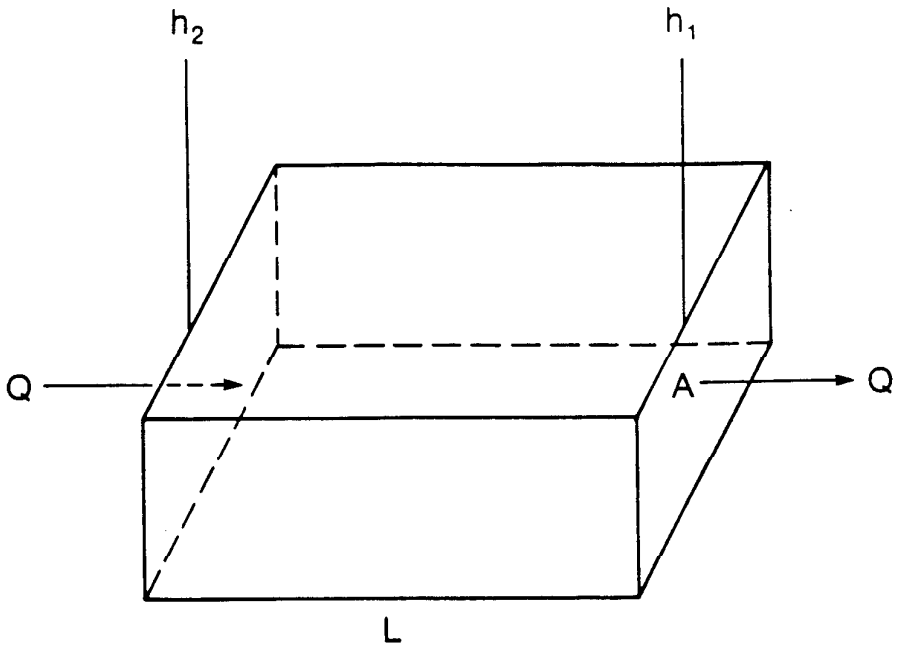
h_2-h_1 is the head differences across the prism parallel to flow (L); and

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.

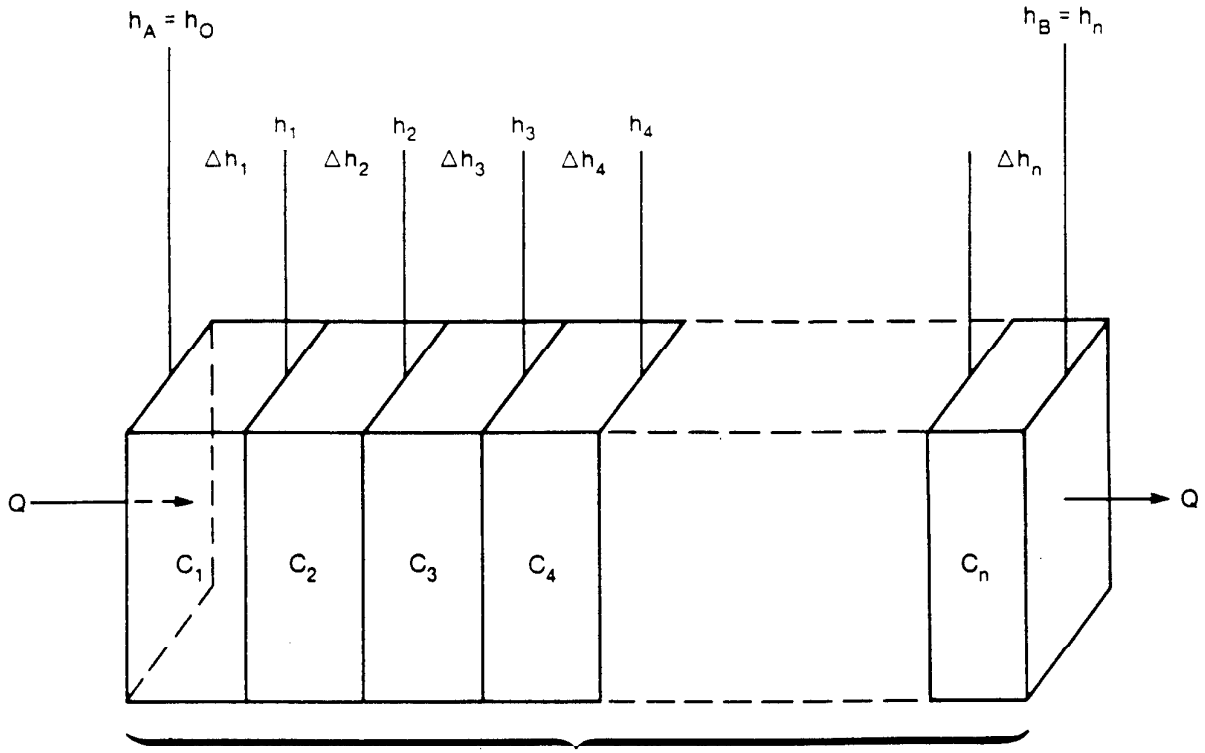
$$C = Q/(h_A-h_B) \tag{34}$$

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,

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

By comparison with equation (34), it can be seen that

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

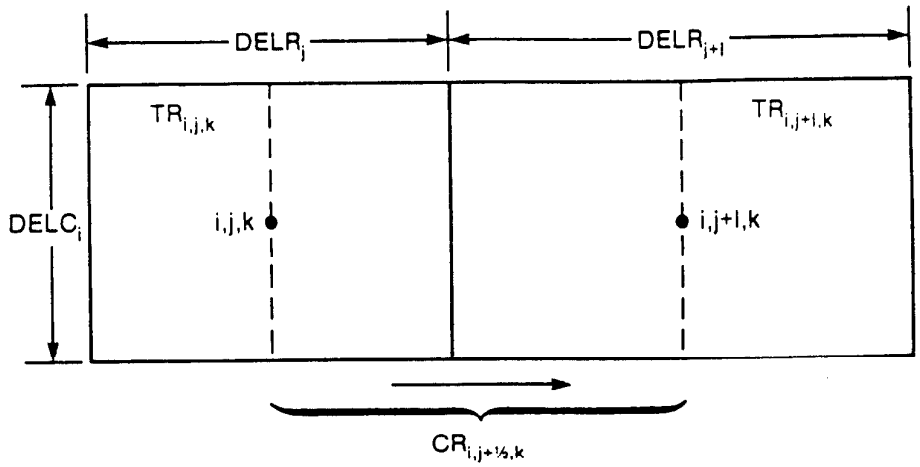
Thus for a set of conductances arranged in series, the inverse of the equivalent conductance equals the sum of the inverses of the individual conductances. When there are only two sections, the equivalent conductance reduces to

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

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

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

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

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

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

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

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

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

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

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

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

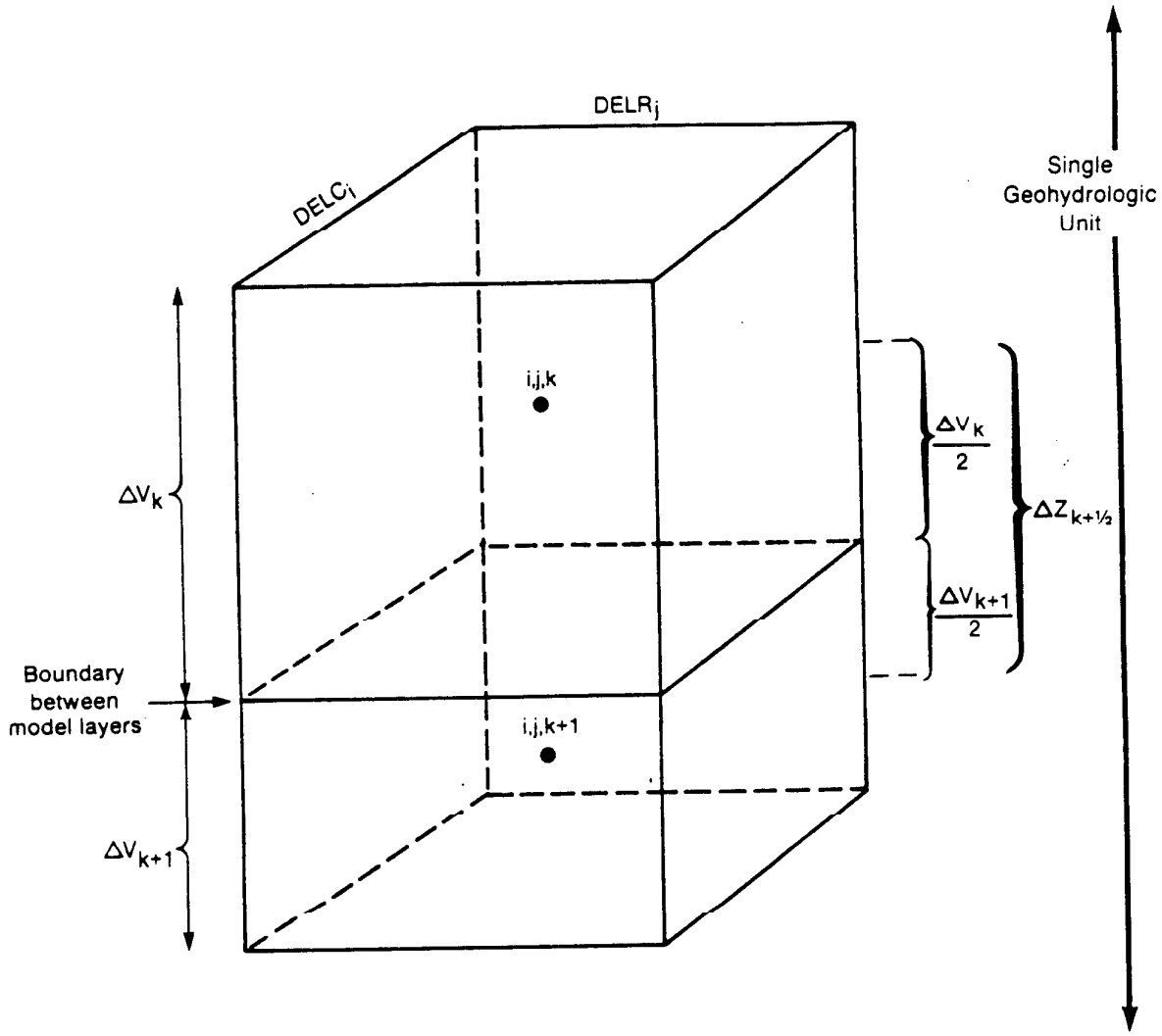


Figure 26.—Diagram for calculation of vertical leakance, Vcont, between two nodes which fall within a single geohydrologic unit.

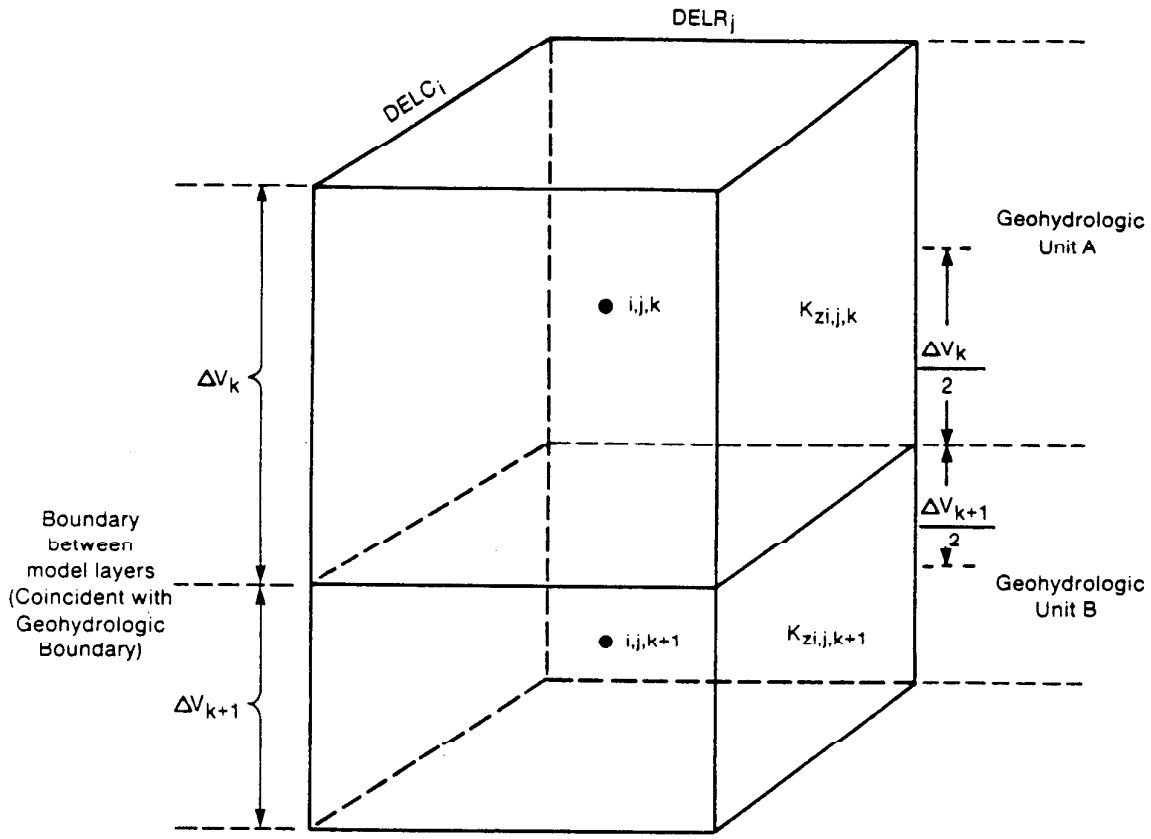


Figure 27.—Diagram for calculation of vertical leakance, Vcont, between two nodes located at the midpoints of vertically adjacent geohydrologic units.

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

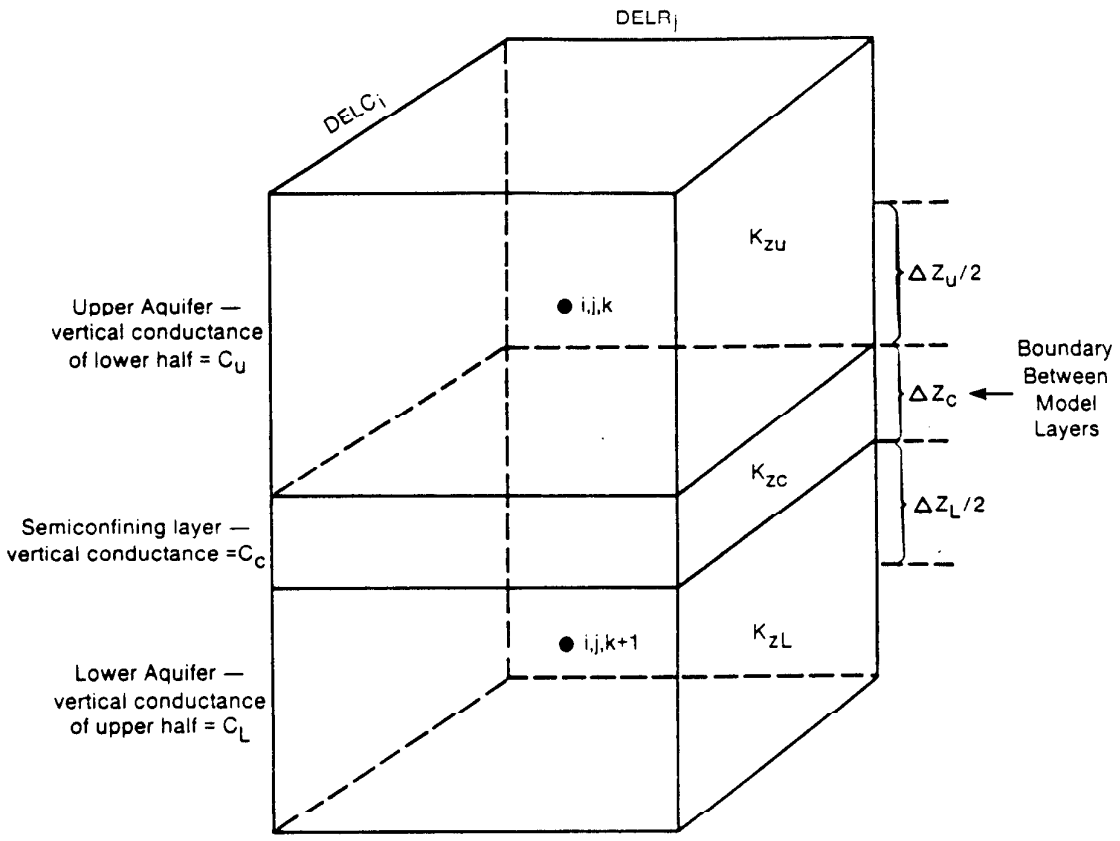
K_{zu} is the vertical hydraulic conductivity of the upper aquifer

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

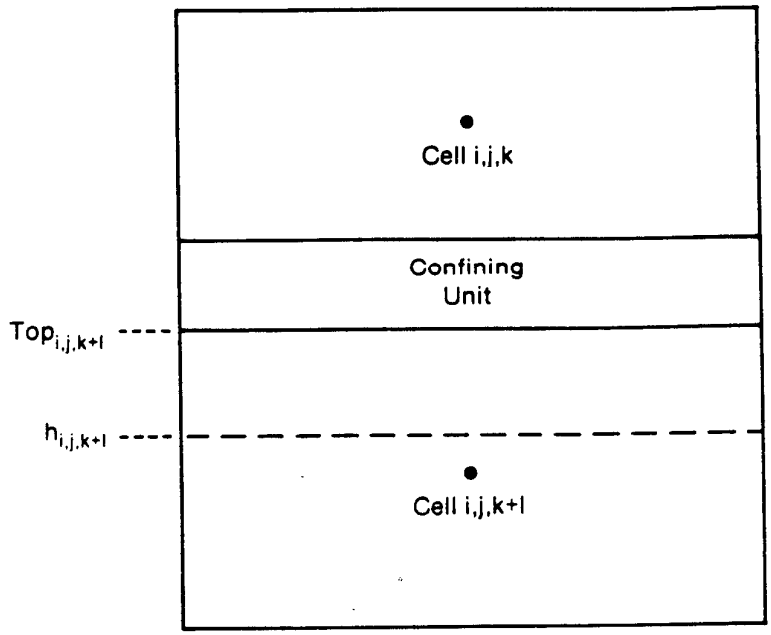


Figure 29.—Situation in which a correction is required to limit the downward flow into cell i,j,k+1, as a result of partial desaturation of the cell.

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 $SCl_{i,j,k}$ was introduced, where $SCl_{i,j,k} = SS_{i,j,k}\Delta r_j \Delta c_i \Delta v_k$. In this report the term $SCl_{i,j,k}$ is termed the "storage capacity" or the "primary storage capacity" of cell i,j,k; the "primary" designation is used to distinguish $SCl_{i,j,k}$ 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

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

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

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

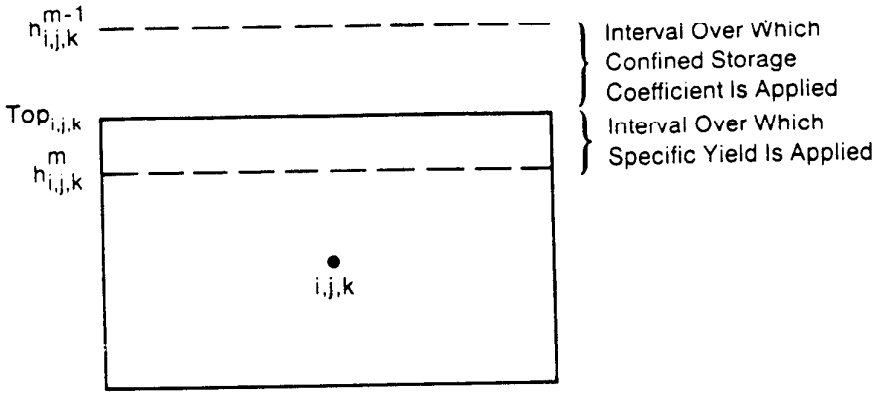


Figure 30.—A model cell which uses two storage factors during one iteration.

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.

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

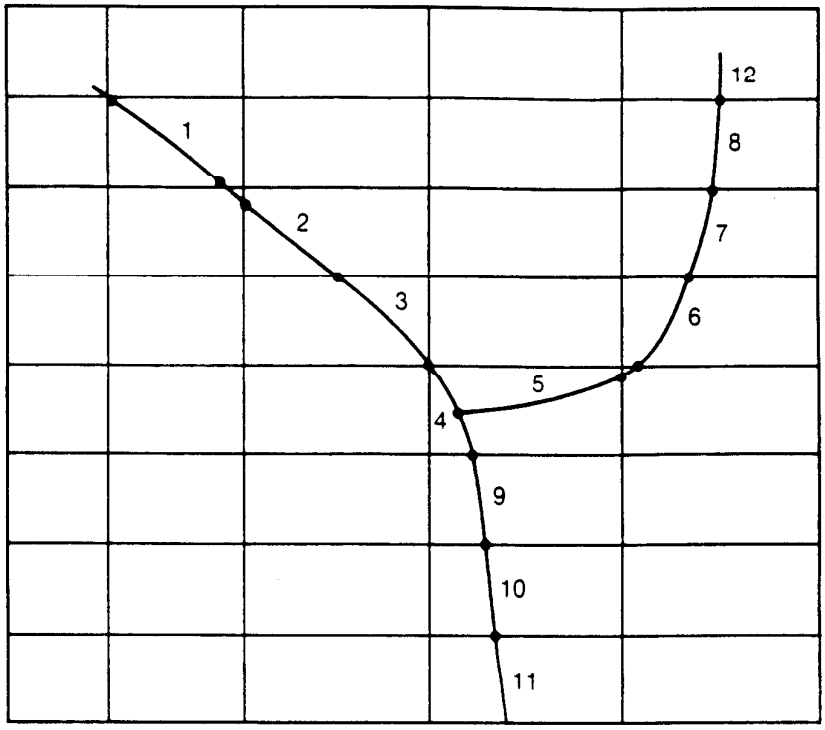
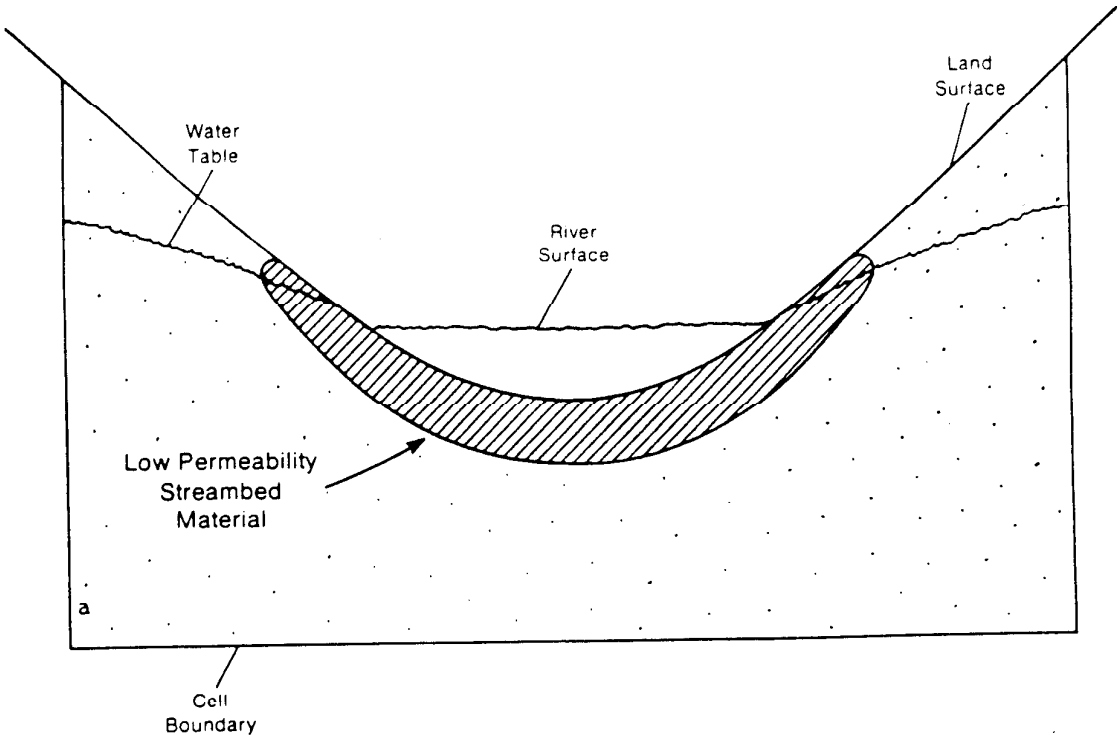


Figure 32.—Discretization of a stream into reaches. Some small reaches are ignored.



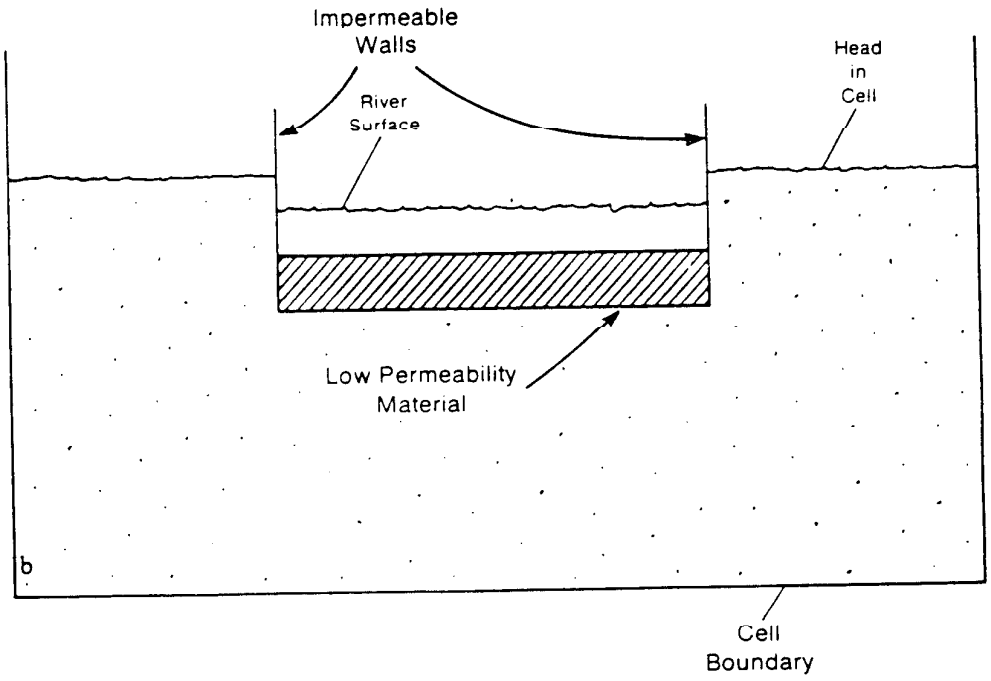
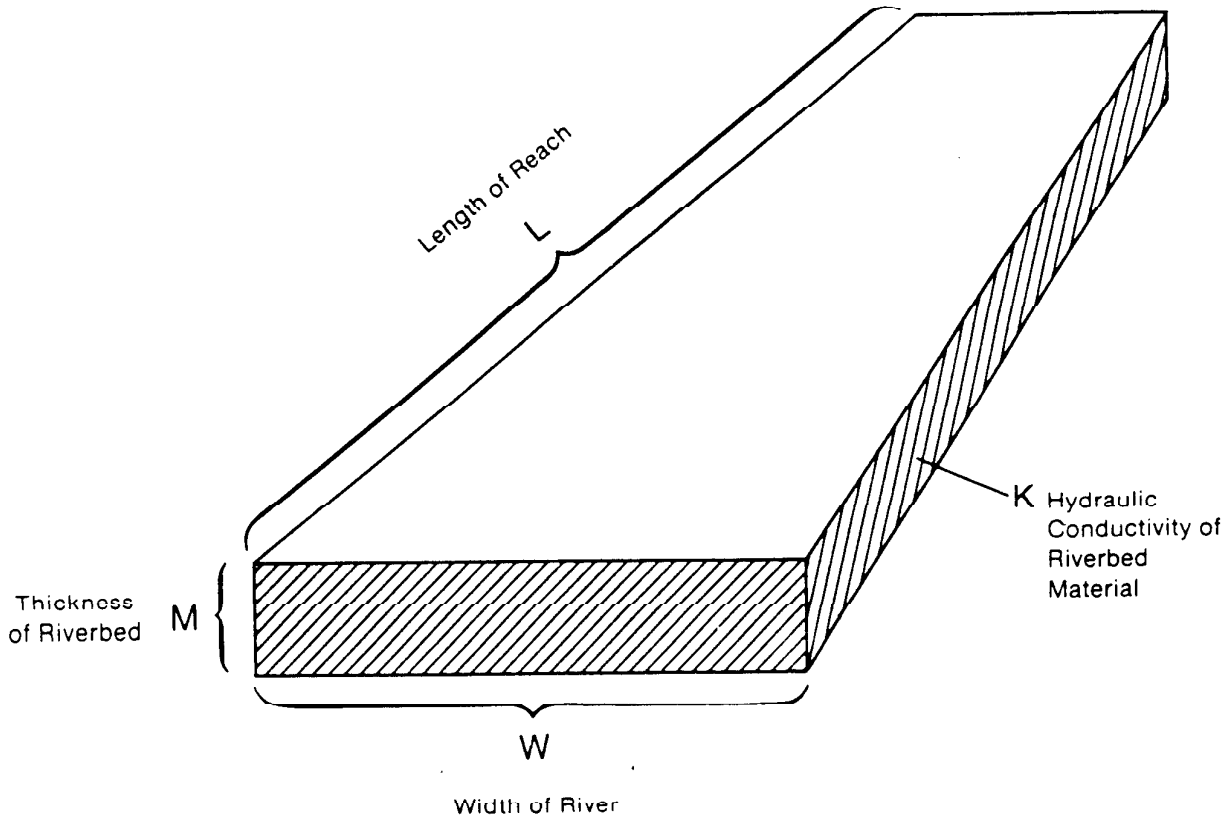


Figure 33.—(a) Cross section of an aquifer containing a stream and (b) Conceptual representation of stream-aquifer interconnection in simulation.



Streambed Conductance = KLW/M

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

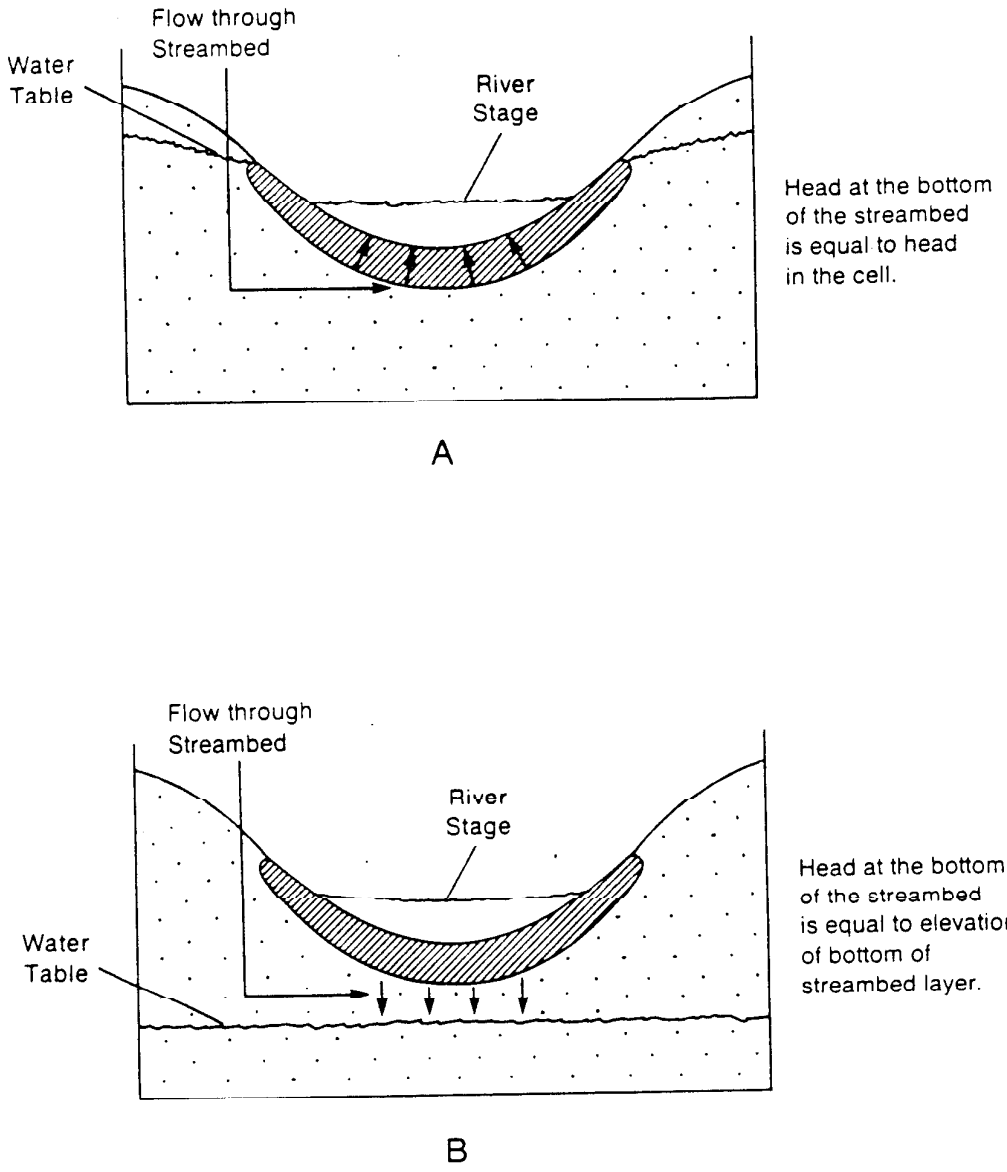


Figure 35.—Cross sections showing the relation between head at the bottom of the streambed layer and head in the cell. Head in the cell is equal to the water-table elevation.

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.

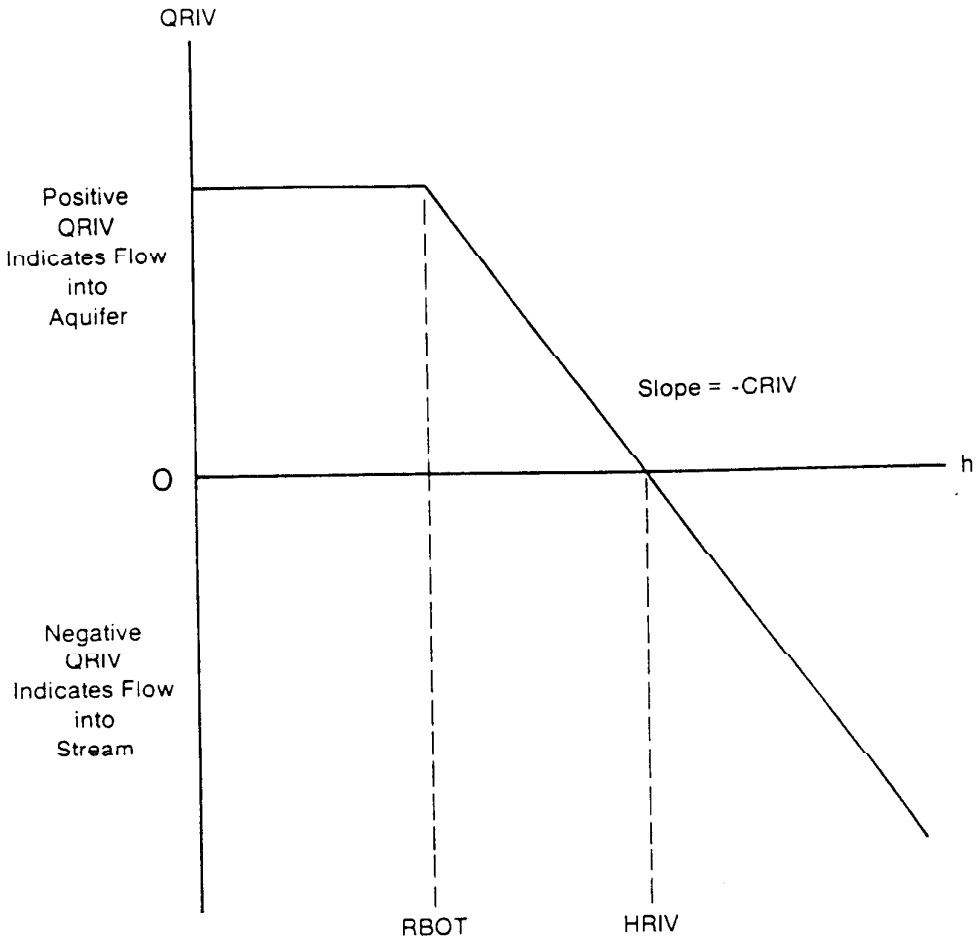


Figure 36.—Plot of flow, QRIV, from a stream into a cell as a function of head, h, in the cell where RBOT is the elevation of the bottom of the streambed and HRIV is the head in the stream.

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

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

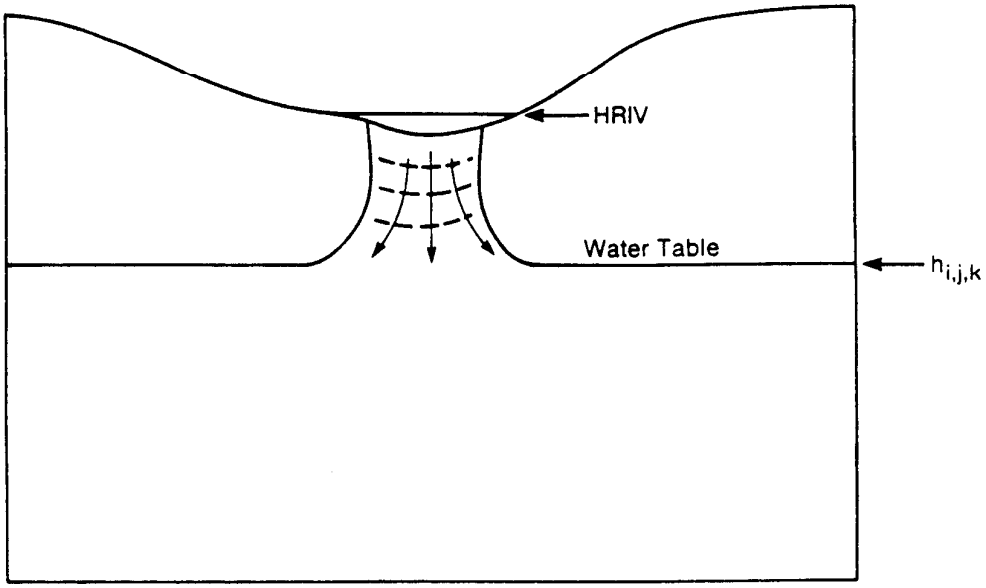


Figure 37.—Limiting seepage from a stream at unit hydraulic gradient

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					

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, $HCOF_{i,j,k}$.

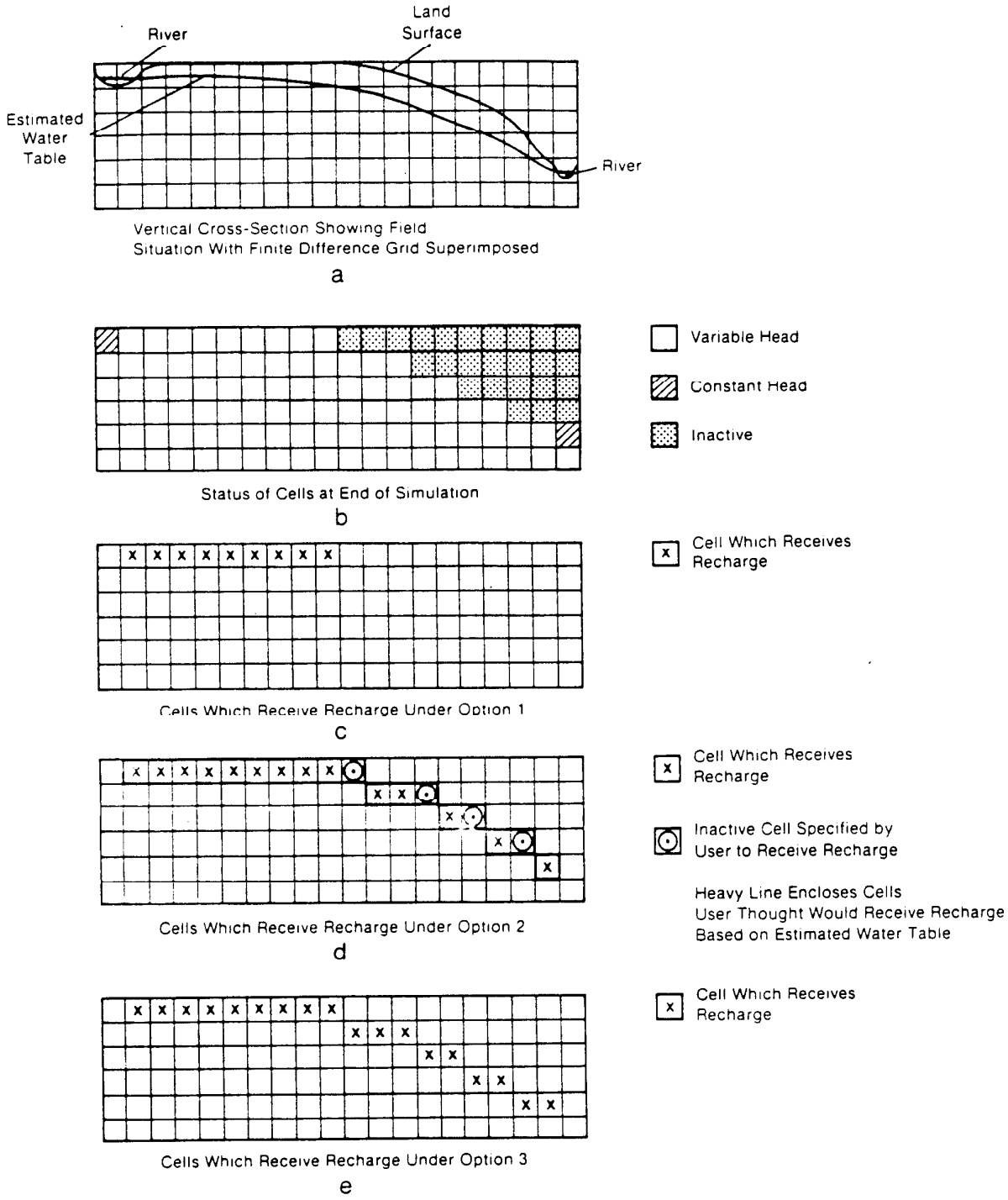


Figure 38.—Hypothetical problem showing which cells receive recharge under the three options available in the Recharge Package.

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)

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

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

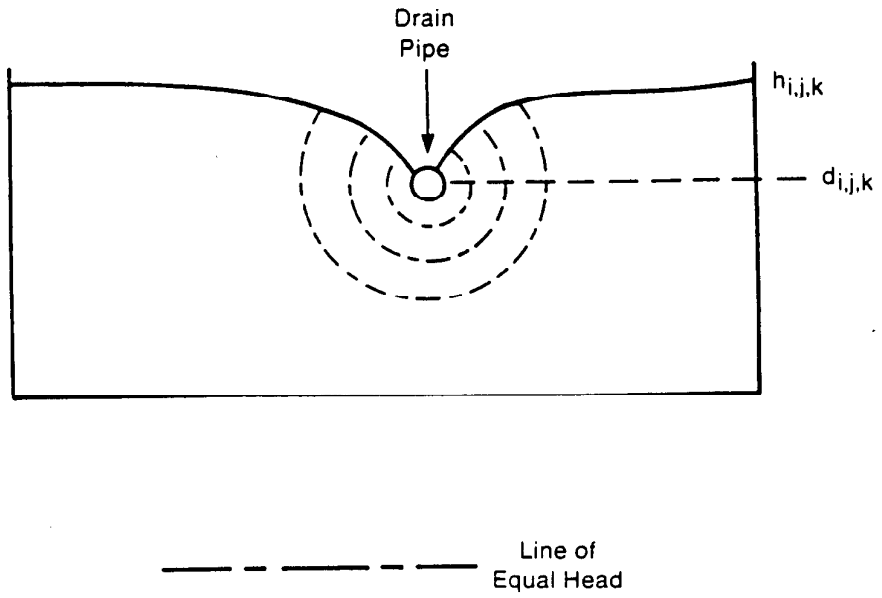


Figure 39.—Cross section through cell i,j,k illustrating head loss in convergent flow into drain.

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

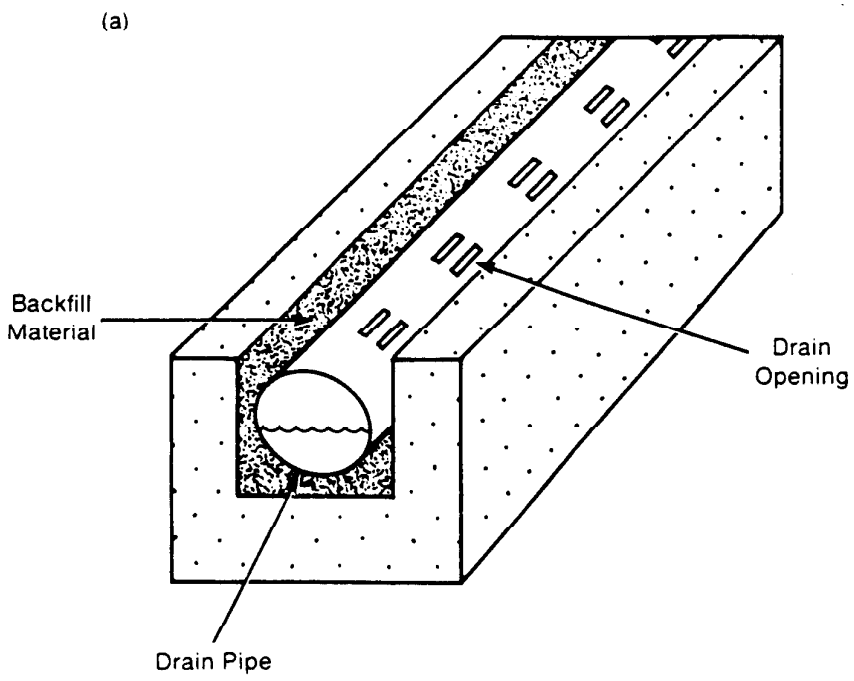


Figure 40.—Factors affecting head loss immediately around a drain: (a) buried drain pipe in backfilled ditch and (b) open drain.

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.

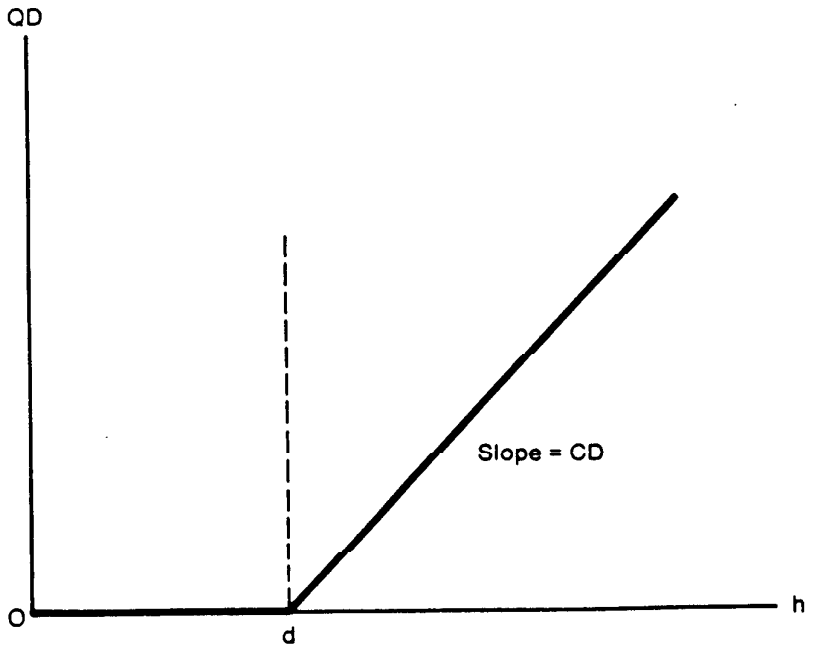


Figure 41.—Plot of flow, QD, into a drain as a function of head, h, in a cell where the elevation of the drain is d and the conductance is CD.

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.

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-

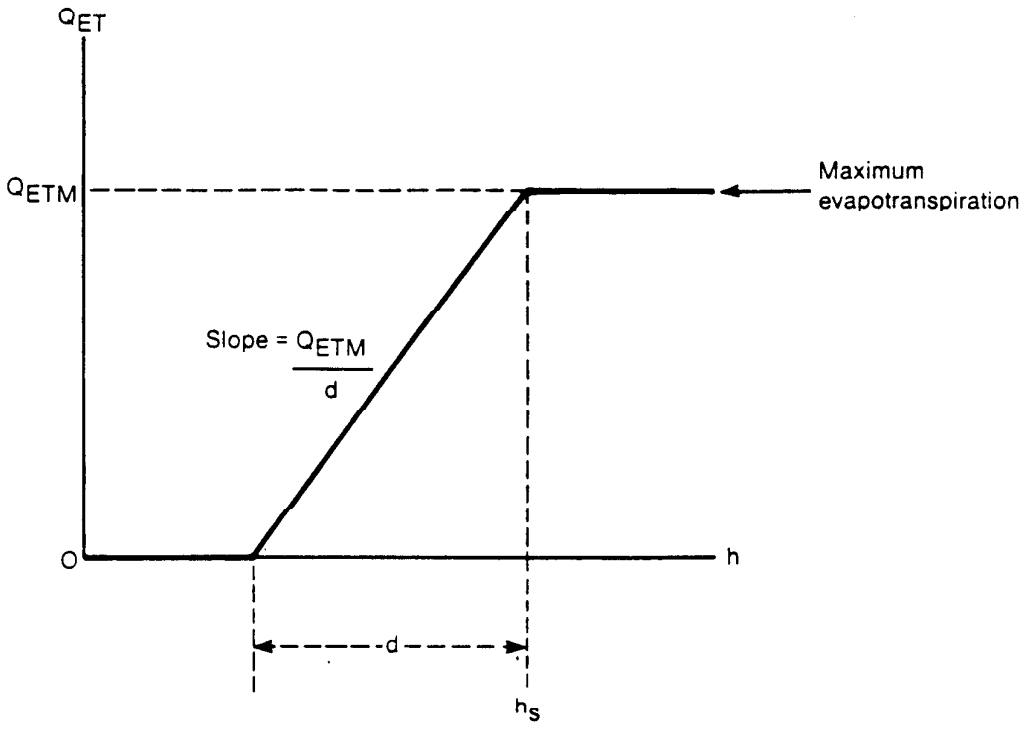


Figure 42.—Plot of volumetric evapotranspiration, Q_{ET}, as a function of head, h, in a cell where d is the cutoff depth and h_s is the ET surface elevation.

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:

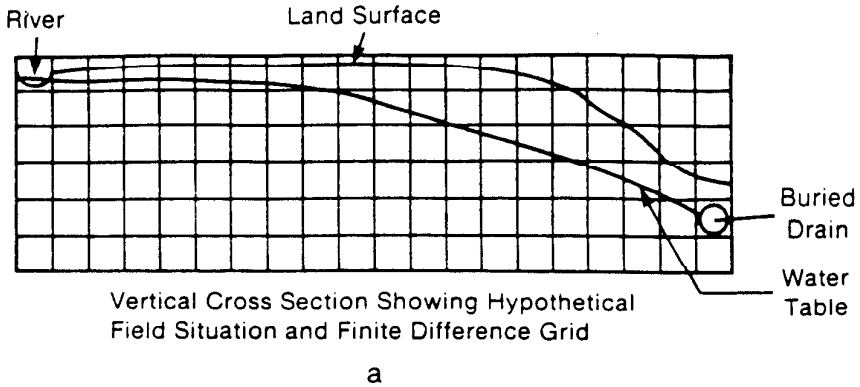
1) if $h_{i,j,k} < (h_{si,j,k} - d_{i,j})$ no changes are made in the terms HCOF or RHS for cell i,j,k;
2) if $h_{i,j,k} > h_{si,j}$, $Q_{ETMi,j}$ is added to $RHS_{i,j}$; and
3) if $(h_{si,j,k} - d_{i,j}) \leq h_{i,j,k} \leq h_{si,j,k}$, $-Q_{ETMi,j}/d_{i,j}$ is added to $HCOF_{i,j,k}$ and $-Q_{ETMi,j}\left\{\frac{h_{si,j} - d_{i,j}}{d_{i,j}}\right\}$ 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.

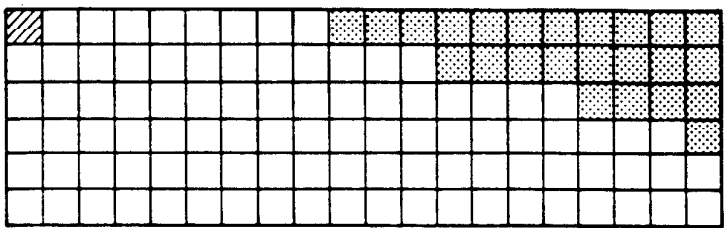
Under option 1 (figure 43-c), evapotranspiration is drawn only from the uppermost layer of the model; in the problem shown, the presence of no flow cells in this layer deletes evapotranspiration from the right half of the model, so that the simulation fails to represent field conditions.

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



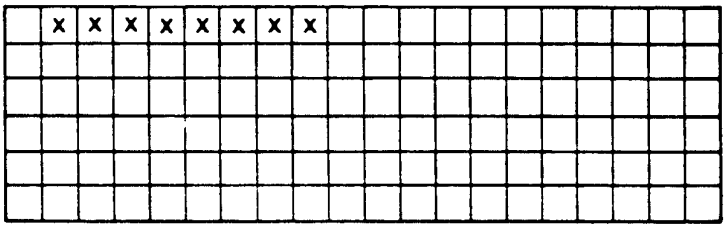
Variable Head Cell

Constant Head Cell

Inactive Cell

Status of Cells at End of Simulated Period

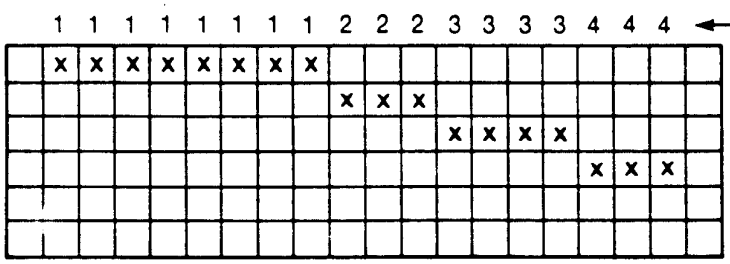
b



Cell from Which ET
Is Abstracted

Cells from Which ET Is Abstracted Under
Option 1

c



Layer Indicators Specified
in the IEVT Array

Cell from Which ET
Is Abstracted

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 REC[illegible]ARE---{ LOCAT, CONST, FMTIN, IPRN}

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

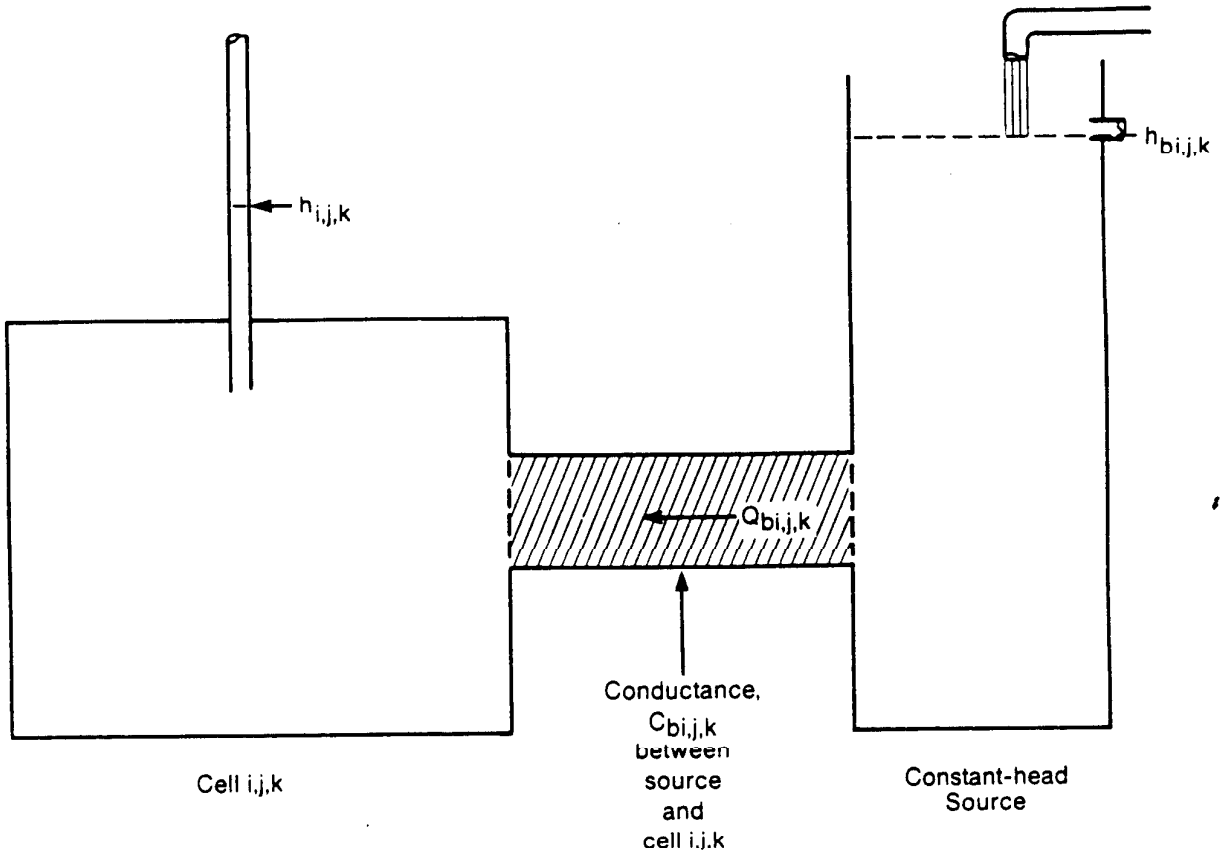


Figure 44.—Schematic diagram illustrating principle of general-head boundary package.

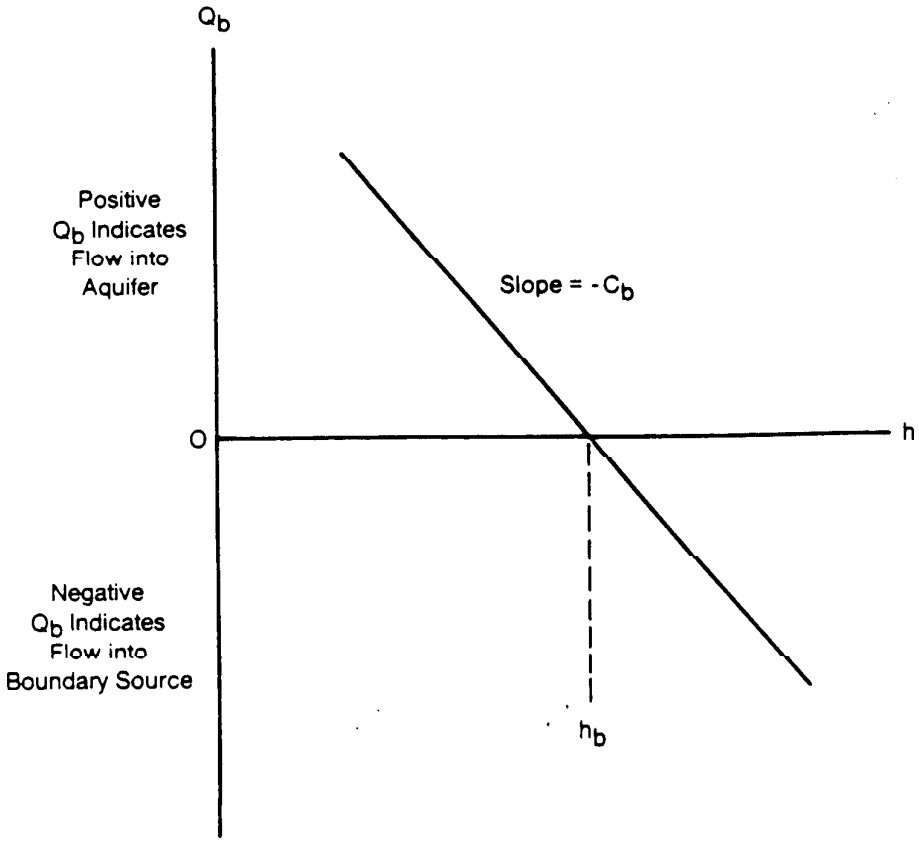


Figure 45.—Plot of flow, Q_b, from a general-head boundary source into a cell as a function of head, h, in the cell where h_b is the source head.

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.

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

$$Z_{i,j,k} = S_{i,j,k-1} \quad (82)$$

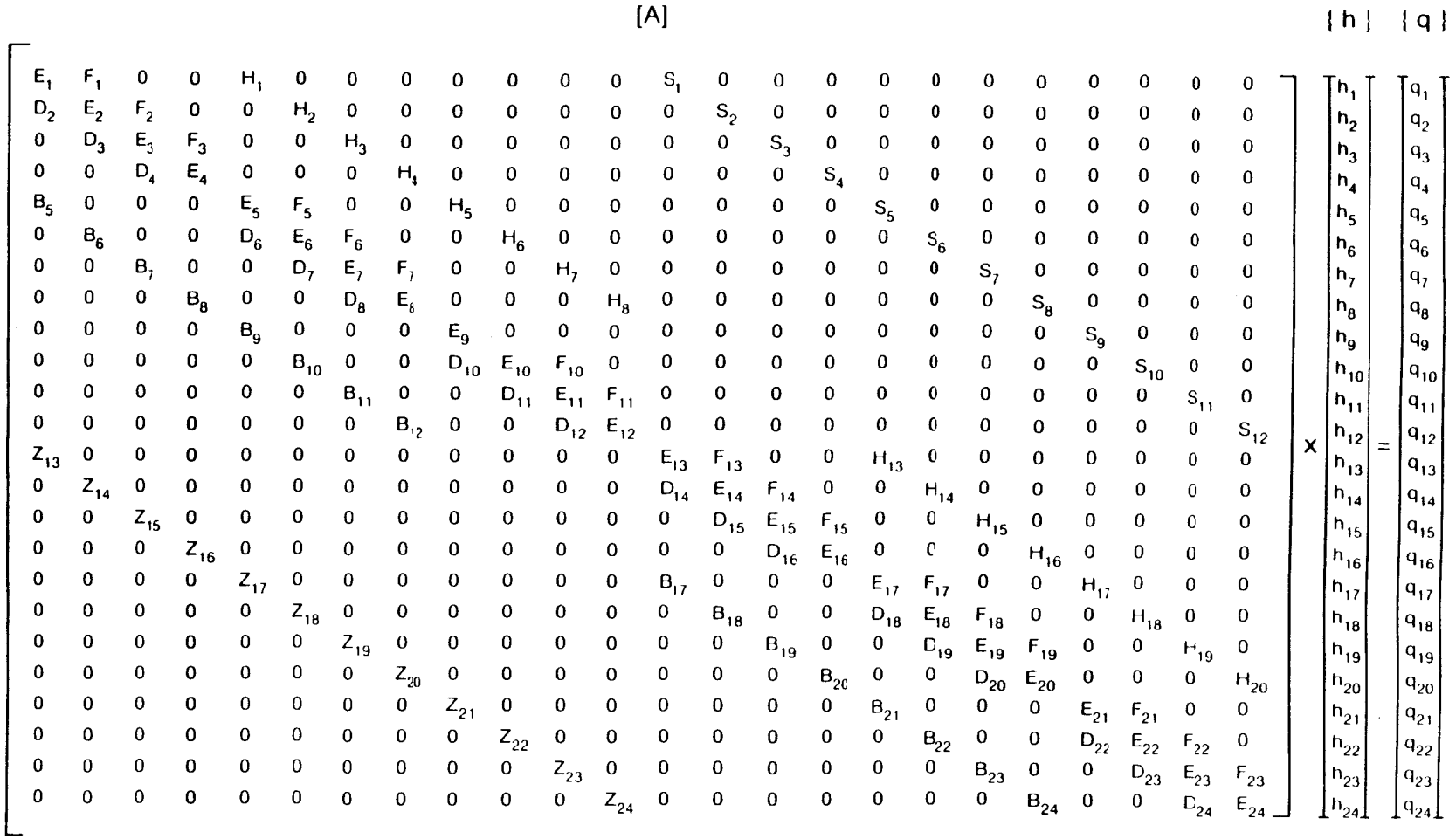
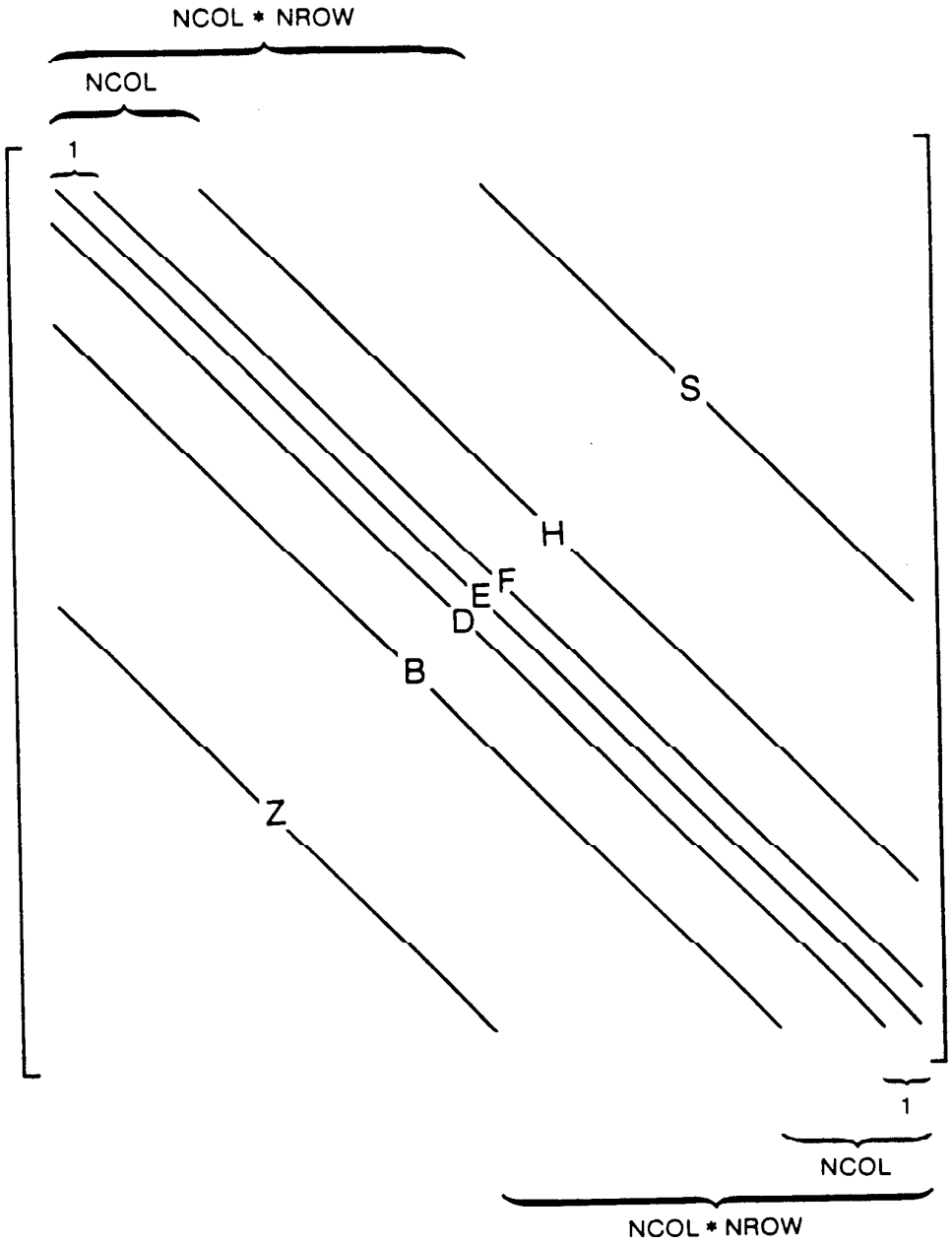


Figure 46.—Correspondence between the finite-difference equations and the matrix equation for a grid of three rows, four columns, and two layers.



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.

$$
\underset{[A]}{\begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 2 \\ 3 & 2 & -2 \end{bmatrix}} \quad \underset{\{h\}}{\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}} = \underset{\{q\}}{\begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}}
$$

$$
\underset{[L^*]}{\begin{bmatrix} 1 & 0 & 0 \\ -1 & 3 & 0 \\ 3 & -4 & -1 \end{bmatrix}} \quad \underset{[U^*]}{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}} \quad \underset{\{h\}}{\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}} = \underset{\{q\}}{\begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}}
$$

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].

$$Z'_{i,j,k} = a_{i,j,k} \tag{89-a}$$

$$A'_{i,j,k} = a_{i,j,k}e_{i,j,k-1} \tag{89-b}$$

$$T'_{i,j,k} = a_{i,j,k}f_{i,j,k-1} \tag{89-c}$$

$$B'_{i,j,k} = b_{i,j,k} \tag{89-d}$$

$$C'_{i,j,k} = e_{i-1,j,k}b_{i,j,k} \tag{89-e}$$

$$D'_{i,j,k} = c_{i,j,k} \tag{89-f}$$

$$E'_{i,j,k} = a_{i,j,k}g_{i,j,k-1} + b_{i,j,k}f_{i-1,j,k} + e_{i,j-1,k}c_{i,j,k} + d_{i,j,k} \tag{89-g}$$

$$F'_{i,j,k} = d_{i,j,k}e_{i,j,k} \tag{89-h}$$

$$G'_{i,j,k} = f_{i,j-1,k}c_{i,j,k} \tag{89-i}$$

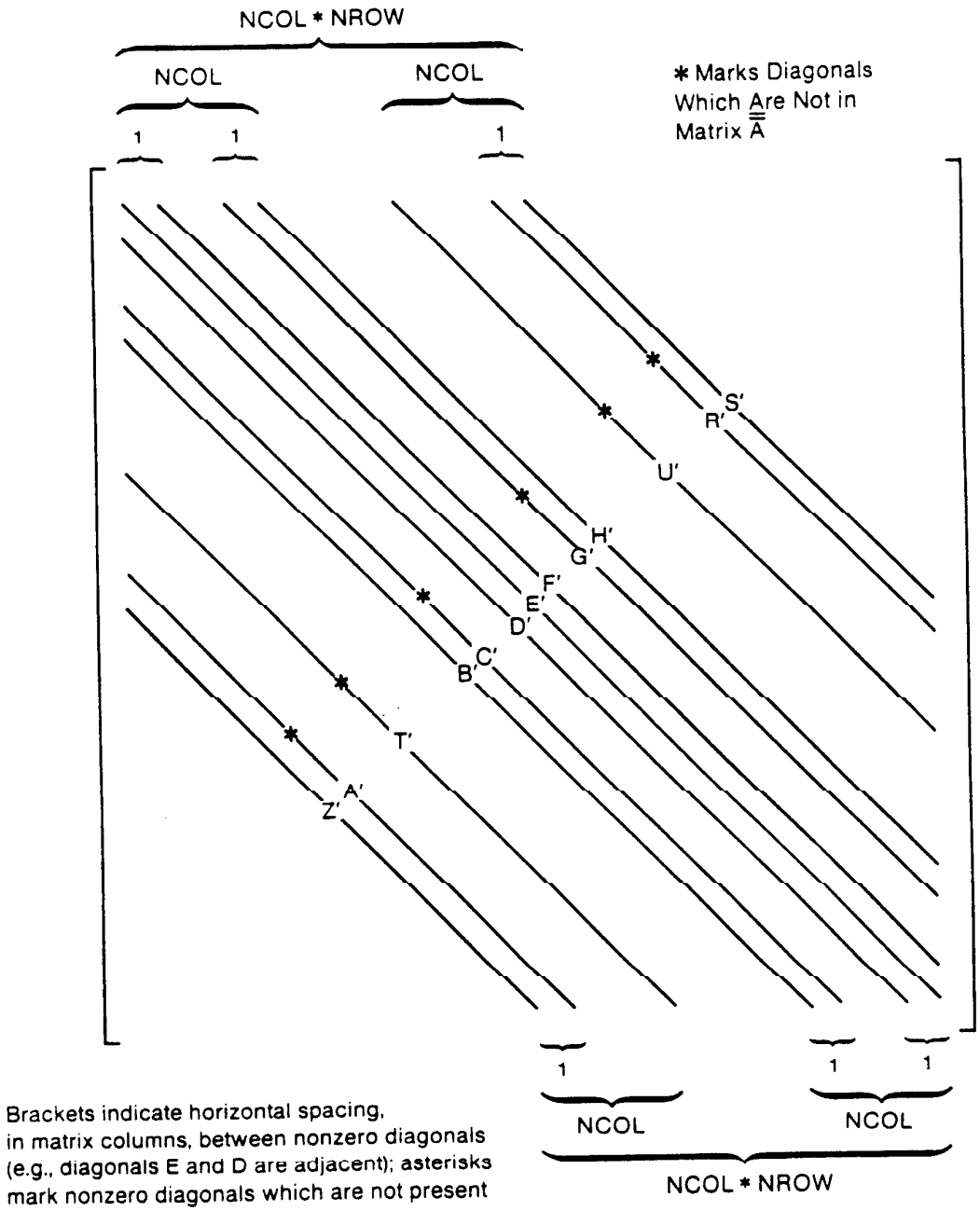
$$H'_{i,j,k} = f_{i,j,k}d_{i,j,k} \tag{89-j}$$

$$U'_{i,j,k} = b_{i,j,k}g_{i-1,j,k} \tag{89-k}$$

$$R'_{i,j,k} = g_{i,j-1,k}c_{i,j,k} \tag{89-l}$$

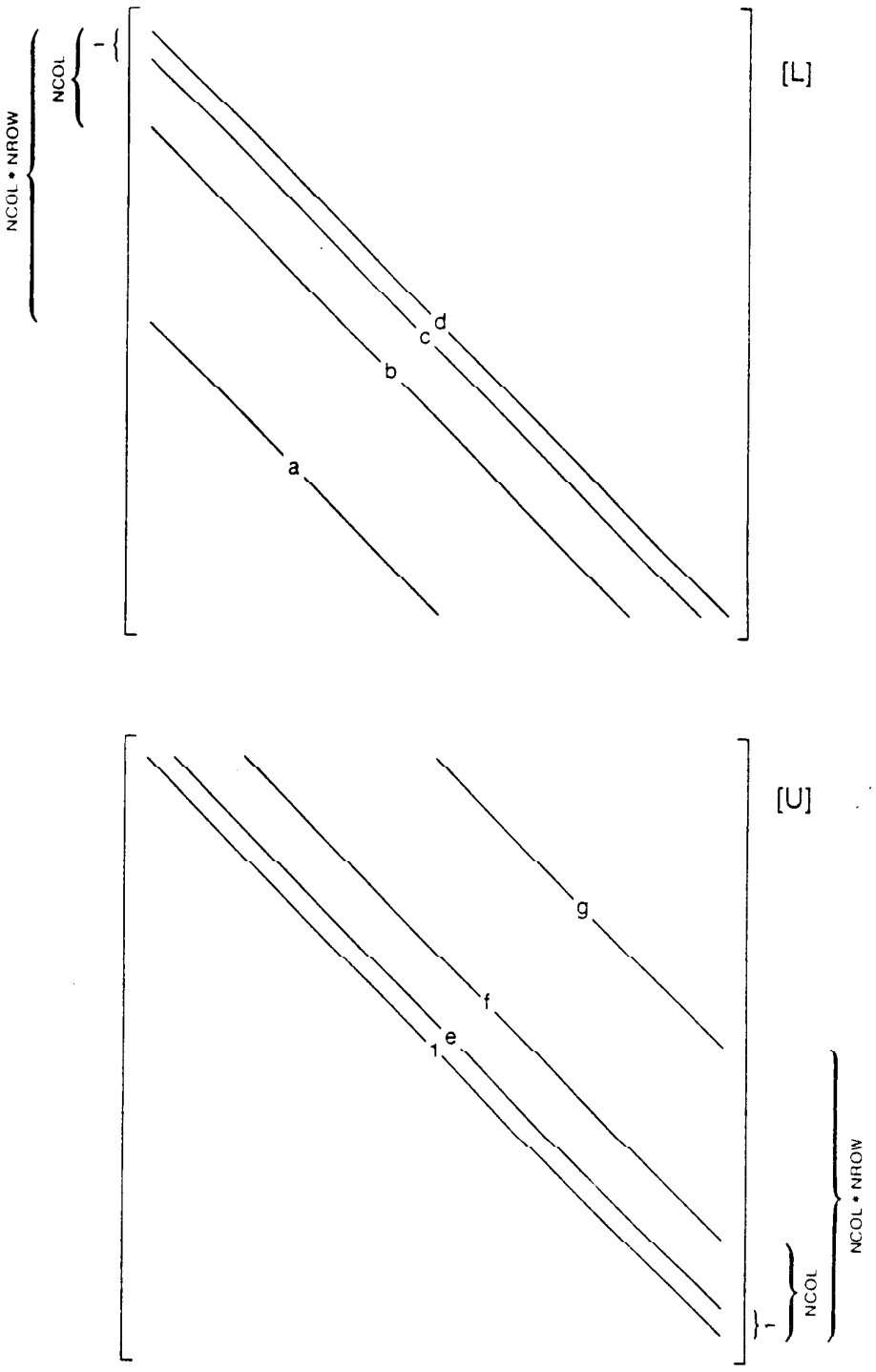
$$S'_{i,j,k} = g_{i,j,k}d_{i,j,k} \tag{89-m}$$

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:

$$
\begin{aligned}
&(Z_{i,j,k}-Z'_{i,j,k})h_{i,j,k-1} + (B_{i,j,k}-B'_{i,j,k})h_{i-1,j,k} \\
&+ (D_{i,j,k}-D'_{i,j,k})h_{i,j-1,k} + (E_{i,j,k}-E'_{i,j,k})h_{i,j,k} \\
&+ (F_{i,j,k}-F'_{i,j,k})h_{i,j+1,k} + (H_{i,j,k}-H'_{i,j,k})h_{i+1,j,k} \\
&+ (S_{i,j,k}-S'_{i,j,k})h_{i,j,k+1} = \alpha A'_{i,j,k}h_{i,j+1,k-1} \\
&+ \beta T'_{i,j,k}h_{i+1,j,k-1} + \gamma C'_{i,j,k}h_{i-1,j+1,k} \\
&+ \gamma G'_{i,j,k}h_{i+1,j-1,k} + \beta U'_{i,j,k}h_{i-1,jk+1} \\
&\alpha R'_{i,j,k}h_{i-1,k+1}
\end{aligned}
\qquad (93)
$$

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

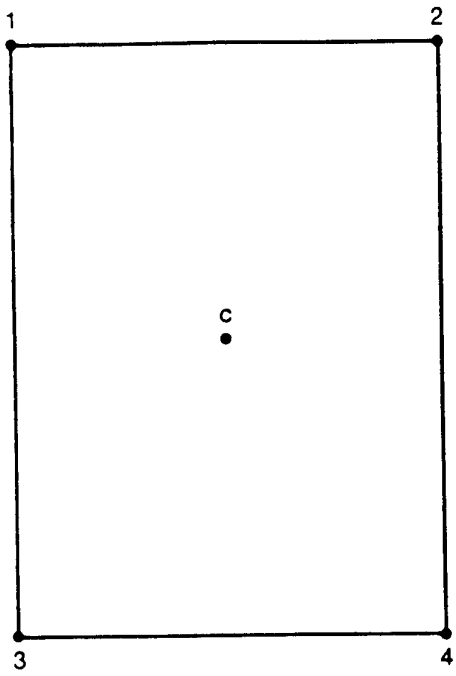
$$h_{i,j+1,k-1} = h_{i,j+1,k} + h_{i,j,k-1} - h_{i,j,k}. \qquad (94\text{-a})$$

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

$$\{RES^{\ell}\} = \{q\} - [A]\{h^{\ell-1}\} \tag{99}$$

Using this notation equation (98) can be written

$$[L][U]\{h^{\ell}-h^{\ell-1}\} = \{RES^{\ell}\} \tag{100}$$

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

$$[L]\{v\} = \{RES^{\ell}\} \tag{101}$$

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

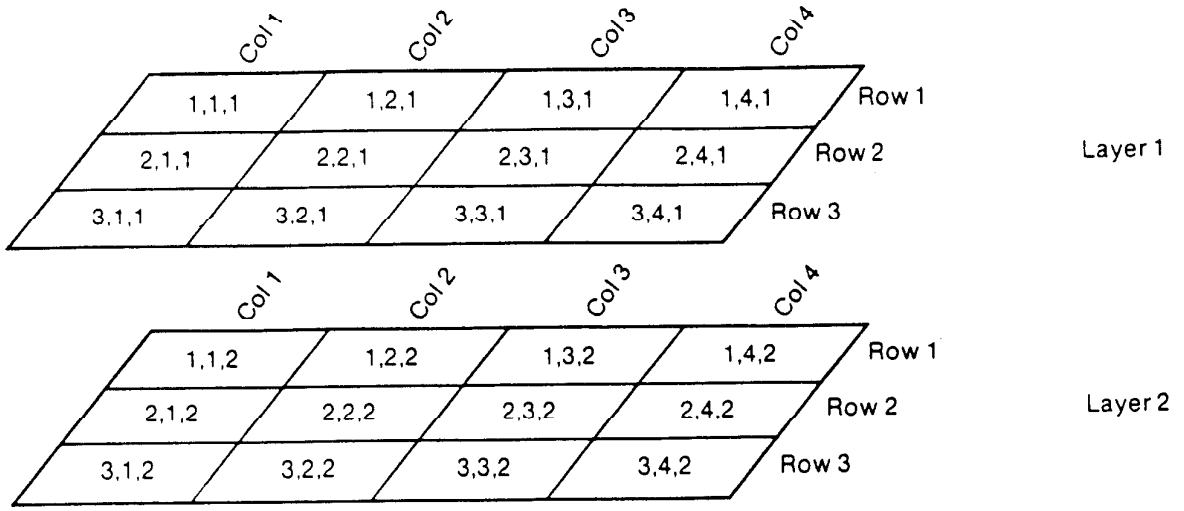
$$[U]\{h^{\ell}-h^{\ell-1}\} = \{v\} \tag{102}$$

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

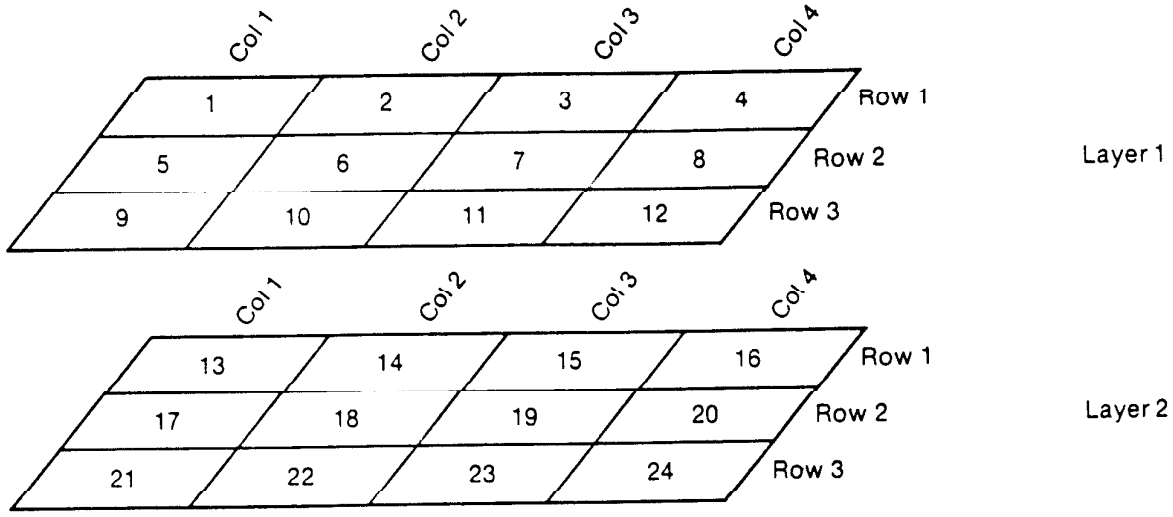
$$d_1 v_1 = RES_1^{\ell} \tag{103}$$

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

$$c_2 v_1 + d_2 v_2 = RES_2^{\ell} \tag{104}$$



(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

$$a_{i,j,k} = Z_{i,j,k}/(1+\alpha e_{i,j,k+1}+\beta f_{i,j,k+1}) \qquad (107\text{-a})$$

$$b_{i,j,k} = B_{i,j,k}/(1+\gamma e_{i+1,j,k}+\beta g_{i+1,j,k}) \qquad (107\text{-b})$$

$$c_{i,j,k} = D_{i,j,k}/(1+\gamma f_{i,j-1,k}+\alpha g_{i,j-1,k}) \qquad (107\text{-c})$$

$$A'_{i,j,k} = a_{i,j,k}e_{i,j,k+1} \qquad (107\text{-d})$$

$$C'_{i,j,k} = e_{i+1,j,k}b_{i,j,k} \qquad (107\text{-e})$$

$$G'_{i,j,k} = f_{i,j-1,k}c_{i,j,k} \qquad (107\text{-f})$$

$$R'_{i,j,k} = g_{i,j-1,k}c_{i,j,k} \qquad (107\text{-g})$$

$$T'_{i,j,k} = a_{i,j,k}f_{i,j,k+1} \qquad (107\text{-h})$$

$$U'_{i,j,k} = b_{i,j,k}g_{i+1,j,k} \qquad (107\text{-i})$$

$$\begin{aligned} d_{i,j,k} = {} & E_{i,j,k} + \alpha A'_{i,j,k}+\beta T'_{i,j,k} \\ & +\gamma C'_{i,j,k}+\gamma G'_{i,j,k}+\beta U'_{i,j,k} \\ & +\alpha R'_{i,j,k}-a_{i,j,k}g_{i,j,k+1}-b_{i,j,k}f_{i+1,j,k} \\ & -e_{i,j-1,k}c_{i,j,k} \end{aligned} \qquad (107\text{-j})$$

$$e_{i,j,k} = (F_{i,j,k} - \alpha A'_{i,j,k} - \gamma C'_{i,j,k})/d_{i,j,k} \quad (107\text{-k})$$

$$f_{i,j,k} = (H_{i,j,k} - \beta T'_{i,j,k} - \gamma G'_{i,j,k})/d_{i,j,k} \quad (107\text{-l})$$

$$g_{i,j,k} = (S_{i,j,k} - \alpha R'_{i,j,k} - \beta U'_{i,j,k})/d_{i,j,k} \quad (107\text{-m})$$

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

$$a_n = Z_n/(1 + \omega(e_{nll} + f_{nll})) \quad (108\text{-a})$$

$$b_n = B_n/(1 + \omega(e_{nrl} + g_{nrl})) \quad (108\text{-b})$$

$$c_n = D_n/(1 + \omega(f_{ncl} + g_{ncl})) \quad (108\text{-c})$$

$$A'_n = a_n e_{nll} \quad (108\text{-d})$$

$$C'_n = b_n e_{nrl} \quad (108\text{-e})$$

$$G'_n = c_n f_{ncl} \quad (108\text{-f})$$

$$R'_n = c_n g_{ncl} \quad (108\text{-g})$$

$$T'_n = a_n f_{nll} \quad (108\text{-h})$$

$$U'_n = b_n g_{nrl} \quad (108\text{-i})$$

$$d_n = E_n + \omega(A'_n + T'_n + C'_n + G'_n + U'_n + R'_n) - a_n g_{nll} - b_n f_{nrl} - c_n e_{ncl} \quad (108\text{-}j)$$

$$e_n = (F_n - \omega(A'_n + C'_n))/d_n \quad (108\text{-}k)$$

$$f_n = (H_n - \omega(T'_n + G'_n))/d_n \quad (108\text{-}l)$$

$$g_n = (S_n - \omega(R'_n + U'_n))/d_n \quad (108\text{-}m)$$

$$v_n = (RES_n - a_n v_{nll} - b_n v_{nrl} - c_n v_{ncl})/d_n \quad (108\text{-}n)$$

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 number 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.

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

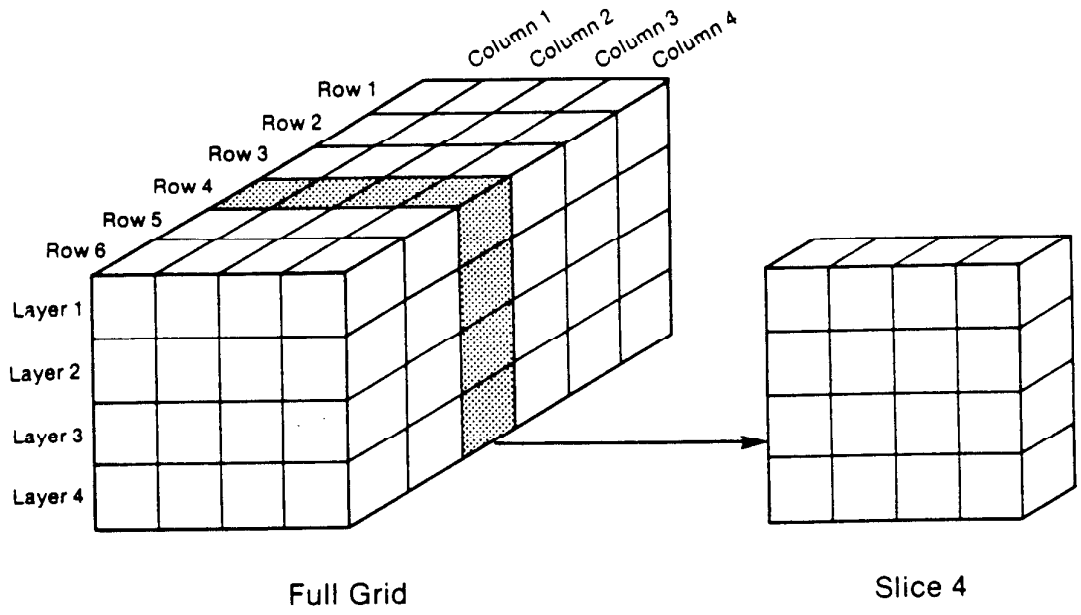


Figure 54.—Division of the three-dimensional model array into vertical slices for processing in the SSOR package.

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

$$
\begin{aligned}
&CV_{i,j,k-1/2}h^{m,\ell}_{i,j,k-1} + CC_{i-1/2,j,k}h^{m,\ell}_{i-1,j,k} + CR_{i,j-1/2,k}h^{m,\ell}_{i,j-1,k} \\
&+ (-CV_{i,j,k-1/2} - CC_{i-1/2,j,k} - CR_{i,j-1/2,k} - CR_{i,j+1/2,k} \\
&- CC_{i+1/2,j,k} - CV_{i,j,k+1/2} + HCOF_{i,j,k})h^{m,\ell}_{i,j,k} + CR_{i,j+1/2,k}h^{m,\ell}_{i,j+1,k} \\
&+ CC_{i+1/2,j,k}h^{m,\ell}_{i+1,j,k} + CV_{i,j,k+1/2}h^{m,\ell}_{i,j,k+1} = RHS_{i,j,k}
\end{aligned}
\tag{113}
$$

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.

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