MICROCOMPUTER CODES FOR SIMULATING TRANSIENT GROUND-WATER FLOW: IN TWO AND THREE SPACE DIMENSIONS

John D. Bredehoeft
U.S. Geological Survey, Menlo Park, California

U.S. GEOLOGICAL SURVEY
Open-File Report 90-559

U.S. DEPARTMENT OF THE INTERIOR
Manuel Lujan, Jr., Secretary

U.S. GEOLOGICAL SURVEY
Dallas L. Peck, Director

For additional information write to:

U.S. Geological Survey
Chief, Branch of Regional Research
345 Middlefield Road, Mail Stop 439
Menlo Park, California 94025

Copies of this report can be purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Federal Center, Building 810
P.O. Box 25425
Denver, Colorado 80225

CONTENTS

ILLUSTRATIONS: Figures:

ABSTRACT

The documentation is presented for a microcomputer code for solving the two dimension and quasi three dimension formulation of the general ground-water flow equation.

PREFACE: GETTING STARTED WITH THE DISKETTE

The source diskette contains both the 2 dimension and the 3 dimension programs. There are two directories on the diskette: XY and XYZ. The two programs are in the two directories: 2D in XY, and 3D in XYZ. Each directory contains the Fortran source program. There are also data sets in both directories which are set-up for test problems; the test problems are further explained in the appendices. In addition there is an executable program for MS-DOS[1] computers designated MAIN; the test problem can be run by simply entering MAIN. The diskette contains all the data necessary to execute the codes.

When the program executes it generates output:

1. The input data files are written to a file GWDATA where they are available following a run.

2. The head change along with the material balance following each designated time step is written to the screen and to a file named GWLEVEL; a final head is also written to this file. GWLEVEL is available after the simulation run.

3. The final head is also written to a file GWSURF. This final head can be used for input to a graphics package or it can be read as an initial head for a continued simulation.

These files require additional disk space. The source diskette containing the two codes is almost entirely full and is write protected. To execute either code the contents of the directories XY or XYZ must be copied to the hard disk or another diskette.

The input files must be changed to simulate different problems. The input files can be edited using the MS-DOS line editor EDLIN[1], or one of any number of word processing programs. The input files are all in a structured format. Care must be taken to preserve the fixed format in editing the input files. One common error is inputing data that violate the read formats of the program; this usually results in a fatal error during execution.

All the data are read by one subroutine LOAD: one can check the read formats in the listing of LOAD in the appendix. Should the input files become too scrambled the test program files can be read again from the source diskette. The codes are compiled and linked to run without a mathematics co-processor chip. The programs will run faster with a co-processor; however, the Fortran source code must be recompiled and relinked for use with a co-processor.

The codes will accommodate large problems; I have run a 7-layer problem with 40 by 30 cells in each layer within the usual 640K memory limitation on a personal computer (PC). A million-year simulation for this problem took several hours to execute in a 286, AT type PC.

1. The use product names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

INTRODUCTION

One of the necessary tools of the ground-water hydrologist is a means to solve the general equation of ground-water flow. With the advent of digital computer space, it is possible to solve the non-homogeneous equations in two or three space dimensions. A number of numerical methods are available for the solution of the general partial differential equation.

Microcomputers have now become available to almost any serious hydrologist throughout the world. With this thought in mind I have set out to develop a relatively simple numerical code to solve the non-homogeneous ground-water flow equations in two and three spatial dimensions. In writing the code there were several objectives:

1. The code should run on a 16 bit computer in single precision; that is, no double precision variables.

2. The code should be written in elementary Fortran 77 so that it is readily transferable between various computers.

3. The code should be readily understandable to anyone knowledgeable with computer programming as well as the basic ground-water flow equations.

4. The code should be designed in such a way that modifications and changes can be readily introduced.

With these objectives in mind, these codes have been modularized into subroutines that perform specific tasks; with the exception of LOAD, subroutines have been kept to between approximately 50 and 100 statements so that their functions are readily understood.

An attempt has been made to relate the code names of variables to names that have meaning to hydrologists. Each of the spatially dimensioned variables is kept in separate, two and three dimensional arrays so that they are readily identifiable with the physical problem.

Because the code is written in FORTRAN 77 it can be run on a variety of small systems, including IBM PC compatible, Macintosh, as well as UNIX based systems. The diskette included with this document contains the FORTRAN source code. It has a compiled, executable program for PC compatible systems. For other systems the source code must be compiled. Some of the file numbers may have to be changed to correspond to the requirements of other operating systems; otherwise there are no significant modifications necessary.

In solving the numerical problem an iterative technique, SIP, which provides good results with all the variables in single precision, has been utilized. There are some rules-of-thumb to keep in mind in solving the numerical problem; some of them are indicated in the sections below. The general idea is to have a relatively simple basic code which one could grasp and utilize on almost any computer with perhaps an hour or so of familiarization.

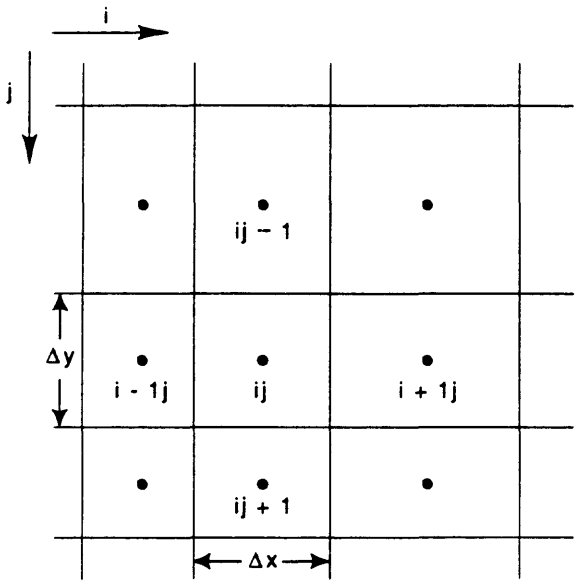

Figure 1. Block centered grid used in the model; the origin is in the upper left hand corner.

NUMERICAL METHODS

The basic flow equation to be solved is:

$$\frac{\partial}{\partial x}(T\frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}(T\frac{\partial h}{\partial y}) = S\frac{\partial h}{\partial t} + \frac{k_c}{l_c}(h - h^{wt}) - W$$

where h is the hydraulic head (L),
T is the transmissivity (L^2/T),
S is the storage coefficient (L^0),
K_c is the hydraulic conductivity of the confining layer (L/T),
(K_c/l_c) is the leakance of the confining layer (1/T),
l_c is the thickness of the confining layer (L),
h^{wt} is head in the overlying source layer (water table) (L), and
W is a source term (-W is a sink) (L/T).

Central finite difference approximations for the spatialderivatives of head are:

$$\frac{\partial h}{\partial x_i} \approx \frac{h_{i+1} - h_{i-1}}{(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}) / 2}$$

$$\frac{\partial^2 h}{\partial x^2} \approx \frac{\dfrac{h_{i+1} - h_i}{(\Delta x_{i+1} + \Delta x_i)/2} - \dfrac{h_i - h_{i-1}}{(\Delta x_i + \Delta x_{i-1})/2}}{\Delta x_i}$$

where $\Delta x_{i-1} = \Delta x_i = \Delta x_{i+1}$

$$\frac{\partial^2 h}{\partial x^2} \approx \frac{h_{i-1} + h_{i+1} - 2h_i}{\Delta x_i^2}$$

A block centered grid with the orgin in the upper left hand corner, as shown in Figure 1, is used. For this code we take as the approximation:

$$\frac{\partial}{\partial x}(T\frac{\partial h}{\partial x}) \approx \frac{(T\frac{\partial h}{\partial x})_{i+1/2} - (T\frac{\partial h}{\partial x})_{i-1/2}}{\Delta x_i}$$

which we will approximate as:

$$(T\frac{\partial h}{\partial x})_{i+1/2} \approx \frac{2T_iT_{i+1}}{(\Delta x_{i=1}T_i) + (\Delta x_iT_{i+1})} (h_{i+1} - h_i)$$

when $\Delta x_i = \delta x_{i+1}$ *then*

$$(T\frac{\partial h}{\partial x})_{i+1/2} \approx \frac{2T_iT_{i+1}}{T_i + T_{i+1}} \frac{(h_{i+1} - h_i)}{\Delta x_i}$$

where $\frac{2T_iT_{i+1}}{T_i + T_{i+1}}$ *is the harmonic mean of adjoining block transmissivities.*

We define:

$$\frac{T_{i-1/2}}{\Delta x_{i-1/2}} = \frac{2T_iT_{i-1}}{\Delta x_{i-1}T_i + \Delta x_iT_{i-1}}$$

We wish to write a finite difference approximation for the flow equation. We will designate the time level: N. Utilizing an iterative method, we designate the iteration level at time N as: IN. (The notation looks formidable; however, the concept is simple.)

It is convenient to define a set of coefficients in both the X and Y directions. In an effort to simplify the notation the indices i and j will be assumed unless modified in some manner:

for example: $h = h_{ij}$; $h_{i+1} = h_{i+1j}$; $\Delta x = \Delta x_i$.

The coefficients are:

$$c^x = \frac{T_{i-1/2}}{\Delta x_{i-1/2}\Delta x}$$

$$c^y = \frac{T_{j-1/2}}{\Delta y_{j-1/2}\Delta y}$$

Two other necessary coefficients are:

$$c^x_{i+1} = \frac{T_{i+1/2}}{\Delta x_{i+1/2}\Delta x}$$

$$c^y_{j+1} = \frac{T_{j+1/2}}{\Delta y_{j+1/2}\Delta y}$$

Using these coefficients we can write an implicit (forward difference) approximation for the general flow equation:

$$c^x h_{i-1}^N + c_{i+1}^x h_{i+1}^N - (c^x + c_{i+1}^x) h^N + c^y h_{j-1}^N + c_{j+1}^y h_{j+1}^N - (c^y + c_{j+1}^y) h^N$$

$$= S \frac{h^N - h^{N-1}}{\Delta t} + \frac{K_c}{l_c}(h^N - h^{wt}) - W$$

where Δt is a finite time step.

Because the iterative solution method, SIP, is used to solve the set of equations an iteration parameter, I, is added to the equation, simply to speed convergence. The iteration term has the form:

$$I(h^{IN} - h^{IN-1})$$

For the purposes of the calculations we compute the change in head for each time step. Then by definition:

$$h^N = H^{N-1} + \epsilon^N$$

$$\epsilon^N = \sum_{IN=1}^{IN-1} \epsilon^{IN} + \epsilon^{IN}$$

$$h^N = h^{N-1} + \sum_{IN=1}^{IN-1} \epsilon^{IN} + \epsilon^{IN}$$

where ϵ^N is the change in head during the time step.

Alternatively we can define the head at anytime, N, as:

$$h^N = h^0 + \sum_{N=1}^{N-1} \epsilon^N + \sum_{IN=1}^{IN-1} \epsilon^{IN} + \epsilon^{IN}$$

where h^0 is the initial head.

The head change exclusive of the current iteration is:

$$R = \sum_{N=1}^{N-1} \epsilon^N + \sum_{IN=1}^{IN-1} \epsilon^{IN}$$

where R is the change in head from time zero, t = 0.

Using these definitions we can rewrite the finite difference approximation to the basic flow equation as:

$$c^x(h^0_{i-1} + R_{i-1} + \epsilon^{IN}_{i-1}) + C^x_{i+1}(h^0_{i+1} + R_{i+1} + \epsilon^{IN}_{i+1}) + c^y(h^0_{j-1} + R_{j-1} + \epsilon^{IN}_{j-1}) + c^y_{j+1}(h^0_{j+1} + R_{j+1} + \epsilon^{IN}_{j+1})$$

$$- (c^x + c^x_{i+1} + c^y + c^y_{j+1})(h^0 + R + \epsilon^{IN})$$

$$= \frac{S}{\Delta t}(h^0 + \sum_{N+1}^{N-1} \epsilon^N + \sum_{IN=1}^{IN-1} \epsilon^{IN} + \epsilon^{IN} - h^0 - \sum_{N=1}^{N-1} \epsilon^N) + \frac{K_c}{l_c}(h^0 + R + \epsilon^{IN} - h^{wt}) + W$$

$$+ I(\sum_{IN=1}^{IN-1} \epsilon^{IN} + \epsilon^{IN} - \sum_{IN=1}^{IN-1} \epsilon^{IN})$$

The right hand side of the equation reduces to:

$$= \frac{S}{\Delta t}(\sum_{IN=1}^{IN-1} \epsilon^{IN} + \epsilon^{IN}) + W + I\epsilon^{IN}$$

The *Strongly Implicit Procedure (SIP)*, introduced by Stone (1968), is used to solve the system of equations. Peaceman (1977) has a full discussion of SIP in two space dimensions. In the SIP method the set of equations is alternately solved in a forward and then a backward direction. The 3D code uses a 3D version of SIP.

UNITS

The code solves the equations independent of units. Any consistent set of units can be utilized. If one selects feet and seconds, for example, then head is specified in feet, pumping in cubic feet per second, transmissivity in feet squared per second, and so forth. Meters and seconds, or feet and days, or meters and days are all acceptable units. I prefer feet and seconds.

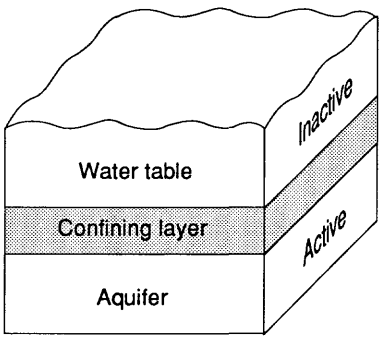

Figure 2. Basic scheme used in the 2D model: an active aquifer, a confining layer, and a source layer (water table aquifer).

BOUNDARY CONDITIONS

The model requires a no-flow boundary around the area of computational interest. Internally the code utilizes the thickness array (THCK) to check on whether a computation is made of not. There must be at least one cell of zero thickness around the entire perimeter within the thickness array. In addition, the model assumes 1) a leaky confining layer, 2) an inactive source layer, and 3) pumping for every cell of interest as shown schematically in Figure 2. Source functions are used to approximate boundary conditions other than no flow.

1. Constant head is achieved in any cell by assigning 1) a high value of leakance along with, 2) a head in the source layer which one would like the aquifer to assume. If the leakance is sufficiently high the aquifer will have approximately the same head as the head in the source layer.

2. Fixed gradient is approximated by putting in a well of given strength to achieve the desired flux at the boundary.

The constant head and fixed gradient boundary conditions (other than no flow, that is, zero gradient) shift the location of the boundary away from the no-flow edge of the cell.

The advantage to this method of approximating boundary conditions is that it is computationally simple; computations for constant head and fixed flux are included in the computations at each cell. The constant head, or fixed flux boundary conditions, can be zero or can take on some fixed value at any cell. The harmonic mean for the interblock transmissivity conveniently sets the coefficient, C, to zero for the no-flow boundary around the area of interest. A schematic for simulating constant head is illustrated in figure 3.

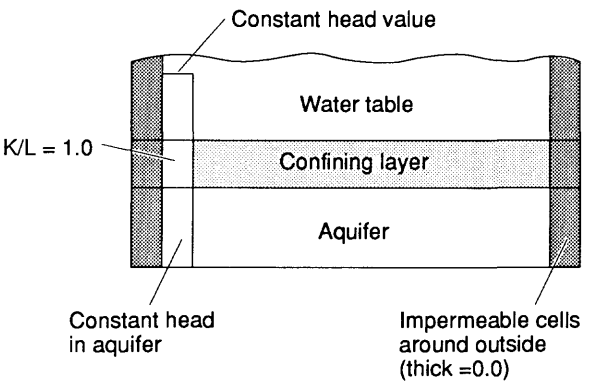

Figure 3. Schematic cross-section utilized to simulate constant head.

QUASI 3D

The basic flow equation for a layered system of multiple aquifers and confining layers is:

$$\frac{\partial}{\partial x}(T_k\frac{\partial h_k}{\partial x}) + \frac{\partial}{\partial y}(T_k\frac{\partial h_k}{\partial y}) = S\frac{\partial h_k}{\partial t} + \frac{K_k}{l_k}(h_k - h_{k-1}) + \frac{K_{k+1}}{l_{k+1}}(h_k - h_{k+1}) + W_k$$

where h_k is the head in the k^{th} aquifer,
T_k is the transmissivity of the k^{th} aquifer,
S is a single storage coefficient used for all layers,
K_k is the hydraulic conductivity of the overlying confining layer,
l_k is the thickness of the overlying confining layer,
K_{k+1} is the hydraulic conductivity of the underlying confining layer,
l_{k+1} is the thickness of the underlying confining layer, and
W_k is a source (or sink) in the K^{th} aquifer.

This is the quasi 3D formulation of the flow equation, which can be derived by integrating, over the vertical, the fully 3D flow equation. The aquifers are coupled through the leakance terms. The basic system of aquifers and confing layers is shown schematically in Figure 4.

This formulation neglects transient flow in the confining layers. Hantush (1960) showed that the effects of storage in the confinig layers could be neglected for large time, where large time was defined as:

$$t_k \geq 10 l_k^2 (\frac{S_k}{K_k})$$

where S_k is the specific storage of the k^{th} confining layer (1/L).

Using an arguement similar to that developed for 2D we can define a finite difference approximation to the quasi 3D flow equation in terms of the change in head (as above the subscripts i, j, and k are assumed, except where modified):

$$c^x\epsilon_{i-1}^{IN} + c_{i+1}^x\epsilon_{i+1}^{IN} + c^y\epsilon_{j-1}^{IN} + c_{j+1}^y\epsilon_{j+1}^{IN}$$

$$- (c^x + c_{i+1}^x + c^y + c_{j+1}^y + \frac{S}{\Delta t} + \frac{K_k}{l_K} + \frac{K_{k+1}}{l_{k+1}} I)\epsilon^{IN}$$

$$= - c^x(h_{i-1}^0 + R_{i-1}) - c_{i+1}^x(h_{i+1}^0 + R_{i+1}) - c^y(h_{j-1}^0 + R_{j-1}) - c_{j+1}^y(h_{j+1}^0 + r_{j+1})$$

$$+ (c^x + c_{i+1}^x + c^y + c_{j+1}^y)(h^0 + R) + \frac{S}{\Delta t}\sum_{IN=1}^{IN-1}\epsilon^{IN}$$

$$+ \frac{K_k}{l_k}(h^0 + R - h_{k-1}^0 - R_{k-1}) + \frac{K_{k+1}}{l_{k+1}}(h_{k+1}^0 + R_{k+1} - h^0 - R) + W_k$$

The 3D code utilizes a three dimensional strongly implicit procedure to solve the basic equations. Three dimensional SIP involves a forward sweep through the matrix followed by a backward sweep (Weinstein, et al, 1969).

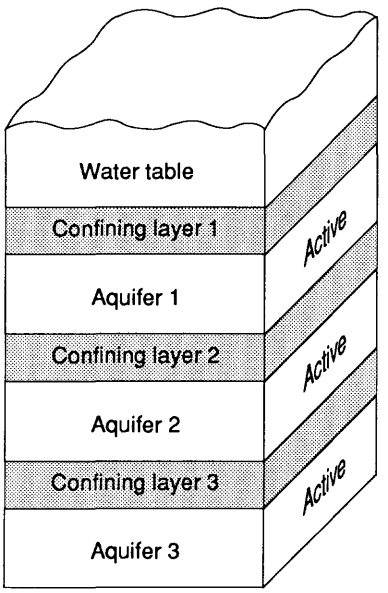

Figure 4. The basic system of aquifers and confining layers assumed for the 3D model.

STRUCTURE OF THE CODE

The code requires a set of files which contain the data input. The necessary files are:

GWPARM
GWINTG
GWTHCK
GWTRAN
GWWTHD
GWHEDI
GWCLTR
GWPUMP
GWXDST
GWYDST

Default values are entered into each of these files. Once they are created on the disk they can be edited. In editing, care must be taken to preserve the fixed format of each file. These data sets are described in detail below.

The model inself consists of a main program and a set of subroutines (subroutines are listed in the order in which they are initially called):

MAIN
Subroutine LOAD
Subroutine COOF
Subroutine ITER
Subroutine SFOR
Subroutine SBAK
Subroutine OUTP
Subroutine MATL

Each of these is discussed in some detail below. The 3D subroutines are the same as the 2D.

MAIN

MAIN is the smallest of the program modules. It controls the logic of calling subroutines. In addition, it steps the program through time. The Subroutine OUTP is called once, initially, so that initial drawdown can be previewed. OUTP is again called after the period of simulation is completed; this ensures an output of the final result.

Steady flow is achieved by setting the storage coefficient, S, equal to O. In the steady-flow situation the program computes two time steps. Comparison of the results of the two time steps should be made to ensure that steady flow has been achieved. A condition code, N = 999, terminates the calculations.

Subroutine LOAD

The Subroutine LOAD calls in the input data from disc storage. This is the largest of the programs and is larger than 100 statements; however, the logic of loading the input data is straightforward. As each data set is called in it is sequentially previewed on the monitor. Pauses are programmed into the subroutine; however, no other manual intervention is currently called for. Pauses can be easily deleted from this subroutine.

LOAD calls in the following data sets:

GWPARM: contains 6 real program variables:

S the storage coefficient (set S = O for steady state).

TOLH the maximum error for convergence.

PERD the period of simulation--years (PERD is multiplied internally by 86400.0 x 365.25).

TIMN the size of initial time step (in units of time selected).

TRMX a transmissivity multiplier.

CLMX a leakance multiplier (not used at cells where CLTR = 1.0).

GWINTG:contains 3 integer program control variables:

NITP is the number of iteration parameters to be used (NITP must not exceed 20, usually 3 or 5 iteration parameters work well).

NPRN the time steps to be printed out (Example: NPRN = 3 prints every third time step).

IRED condition code to read file GWSURF, the initial head (IRED = 1 reads GWSURF as initial head HI ( ) rather than GWHEDI).

GWTHCK: Aquifer thickness array. The thickness is used throughout the program as the principal check to control the computations. There must be at least a border of one cell around the entire perimeter of this array set to 0.0. Any real positive value is acceptable for the aquifer thickness. The aquifer geometry is adjusted using the thickness array; no computations are not made for cells where the thickness is equal to 0.0. For artesian computations it is often convenient to set the thickness for active cells equal to 1.0.

GWTRAN:Transmissivity array. The program is set up so that transmissivity is the input expected. In the case of water-table calculations the input expected is the initial transmissivity (not hydraulic conductivity).

GWCLTR:Leakance array. These values should be leakance layer permeability divided by the thickness of the leakance layer. This array is also used to achieve constant head; a value of 1.0 for CLTR is a flag to the code that a head approximately equal to the source layer head is to be set at the particular cell; that is, a constant head boundary. At constant head cells the code calculates a leakance value 10 times the value of transmissibility divided by the cell area, which achieves a head approximately equal to that in the source layer.

GWHEDI: Initial head array. This is the initial head within the active layer unless GWSURF is read.

GWSURF: Initial head array generated by a previous model run that can be used as a new starting condition.

GWWTHD: Source layer head array. This is the head within the inactive source layer. Assigned value in this array combined within a leakance value of 1.0 (CLTR = 1.0) will approximate the assigned constant head in the aquifer.

GWPUMP: Array of potential pumping. The pumping is designated as volume per unit time, for example, cubic feet per second (ft^3/sec).

GWXDST: Array of cell dimensions, delta x values. This array must have values for each active cell. The values can change from cell to cell. A good rule-of-thumb is not to change cell dimensions by more than a factor of 2 or 3 in adjacent cells.

GWYDST: Array of cell dimensions, delta y values; comparable to XDST.

Once this data is loaded the program checks the thickness matrix, THCK, and sets the pertinent values to 0.0 in each cell where the THCK = 0.0. The Subroutine then initializes the head matrices used in the computations: HD, HR and HC. The input data files are available following a model run.

Subroutine COOF

Subroutine COOF calculates a coefficient matrix utilizing the harmonic mean and stores these in the three dimensional array COEF. The coefficients are calculated as:

$$C_{i,j,1} = 2T_{i-1}T / \{\Delta x_{i-1}T + \Delta xT_{i-1}\}$$

and

$$C_{i,j,2} = 2T_{j-1}T / \{\Delta y_{j-1}T + \Delta yT_{j-1}\}$$

There are two coefficients at each cell. For the 3D code the third index variable, k, is dimensioned to 2 times the number of layers; for example, for 3 layers, k = 6.

Once these values are entered into COEF, the COEF array is examined to determine the minimum iteration parameter. The procedure is one suggested by Stone (1968) and described by Peaceman (1977, p. 134). After selecting the minimum parameter, a geometric sequence of NITP iteration parameters, stored in AOPT (up to 20), is created. Finally, the complete set of iteration parameters is displayed on the screen, consisting of the values in the array AOPT (20); only the entries specified by NITP will have non-zero values. AOPT must be less than, or equal to, 20. Usually an odd number of iteration parameters often 3 or 5, work well.

Subroutine ITER

The Subroutine ITER controls the iterations for any given time step. The first operation is to increment the time. The program is set up to increase the size of the time steps in the following way:

$$\Delta t^N = 1.5\ \Delta t^{N-1}$$

A check is made to adjust the last time step such that the period of simulation will exactly equal PERD (It should be remembered that for convenience PERD is multiplied internally by 864000.0 x 365.25; if the unit of time is seconds, then PERD is specified in years).

The subroutine then begins the iterations. With the SIP procedure, used Subroutine SFOR and SBAK are called in sequence. If the maximum head change computed for the iteration is less than TOLH then the solution is considered to have converged; ITST is set to 1 and the iterations are terminated:

$$\text{MAX} \mid \text{HC(i,j)} \mid < \text{TOLH}$$

Several write statements are displayed during the iterative procedure. These statements indicate the number of the iteration being computed, a statement when the backward, SBAK computations are complete, and the value of maximum head change computed:

$$\text{MAX} \mid \text{HC(i,j)} \mid$$

The maximum number of iterations is currently set to 50. If 50 iterations are completed the program does not terminate, but simply moves on to the next time step. If the iterations equal 50 then one must be aware that the solution has not converged within the tolerance, TOLH, specified. Often this is not a fatal condition; however, one must be careful in utilizing such results.

Finally, the Subroutine ITER checks to see if an output is called for. An output is called for after any iteration divisable by NPRN; i.e., NPRN = 3, every third time step.

Subroutine SFOR

Subroutine SFOR makes the forward, increasing i and j computations in the SIP procedure; for a complete description see Peacement (1977, p. 132-134). Intermediate values are stored in a temporary set of 3 arrays: E ( ), F ( ), V ( ). These arrays also limit the problem size; they must be increased along with the other arrays in order to increase the problem dimensions.

Subroutine SBAK

Subroutine SBAK is similar to SFOR except that it makes implicit backward, decreasing j, computations in the SIP procedure. In the 3d code the backward sweep is made by decreasing both k and j. Following the computations the change in head computed by SFOR (or SBAK), the absolute values of HC (i, j), is compared in each cell to see if convergence has been achieved:

$$\text{MAX} \mid HC(i,j) \mid < TOLH$$

If a head change is greater than TOLH, a condition code, ITST, is set to 0, indicating an additional iteration is necessary.

Subroutine OUTP

The Subroutine OUTP is designed to display the results. In the present configuration of OUTP, results are displayed on the users monitor and written to a file, GWLEVEL, on the disc. GWLEVEL can be listed following the completion of the simulation. A second file GWSURF is also written to the disc; the cell location (i,j or i,j,k) and the final head, HD( ), are written to this file. GWSURF can be read as the initial head for a continuing simulation.

An initial and a final output are created by the MAIN program. An additional output is generated every NPRNth time step. If NPRN is given a sufficiently large value, only an initial and a final output will be created.

Subroutine MATL

The material balance sums 1) the total pumping volume, 2) the total water removed from storage, and 3) the rates of leakance in, leakance out, the rate of change of storage, and the rate of pumping. This allows one to compute a total material balance, and to examine the rates of flow in the system at any time. Our experience suggests that for problems without leakance (when CLTR = 0.0) the total volume from storage will be equal to the total water pumped to 3 or 4 significant figures. This is some indcation of accuracy of the solution, even with all the computations and variables in single precision on a 16-bit machine.

Errors in the material balance are generally introduced by the method for handling the constant head boundaries. The difference between:

$$WTHD(i,j) - HD(i,j)$$

must be sufficiently large such that it is significant for the computations, and yet this difference should not be so large as to lose the effect of the constant head boundary. This fact makes selecting the appropriate leakance value, CLTR (i,j), at the constant head boundaries critical to the material balance computations used in the program. The leakance value at designated constant head nodes is generated internally in Subroutine LOAD.

MODELLING REAL PROBLEMS (Hints)

One cannot teach ground-water modeling in the documentation for particular codes such as these. The art of modeling a real system is to be able to abstract that system into a surrogate that can be analyzed without losing the essence of the real system. It is the skill of the hydrogeologist which makes a person either a good or a poor modeler. Usually the skill of modeling is something one gains through experience. There are, however, a few suggestions that can prove helpful. Often these ideas revolve around how one handles either: 1) the sources of sinks, and/or 2) the model boundaries. Often it is the constant head or the constant flux boundaries which are the biggest problems.

Distributed Recharge

Distributed recharge such as might result from rainfall over the entire area is usually best handled in this code as positive pumping (injection) distributed over the entire area. One should remember that pumping is entered into the model as a volume per unit time. Rainfall over entire cell area would be the appropriate volume to enter as the pumping volume; this would be divided by the appropriate time to obtain a rate. Recharge is a positive pumping in the model.

Streamflow

Streamflow can be handled as leakance out of (or into) the model aquifer to an overlying constant head source layer with cells established along the river. The head in the overlying layer should be set to the river stage. The total leakance both in and out of aquifer is given in the material balance. Assuming there are no other non zero constant head cells in a model, the leakance to these nodes would be the base flow to the stream from the modeled reach of aquifer.

Recharge and Discharge as Leakance

Often the natural driving force for the confined aquifer system is the available topographic relief in the basin. Flow through the aquifer is limited by two factors: 1) the transmissivity of the aquifer, and 2) the available topographic relief as the driving force. In such systems one modeling procedure is to assume that the water table forms the top surface of the saturated groundwater system and that the water table is a subdued replica of the topography. The topography can be entered into the model as the head in the source layer (WTHD). The confined aquifer is then coupled to the inactive source layer through the leakance. The amount of recharge (and/or discharge) will be established by: 1) the transmissivity of the aquifer, and 2) the hydraulic gradients. The aquifer's ability to transmit water will limit the recharge (as well as the discharge). This point is important; another way to state this is: one does not have to specify explicitly recharge (or discharge); one can let the model establish these quantities.

This method of handling the boundaries has one very real pitfall; there is no limit to the amount of flow (recharge or discharge) than can occur at a constant head boundary. One can generate very unrealistic flows at such boundaries, especially if there are stresses in the aquifer inducing large quantities of recharge. Care should be taken to check the flows at such boundaries to ensure that they are reasonable.

Head at a Pumping Well

The head calculated in any block can be viewed as an average head within that block; this includes blocks which contain pumping. Peaceman (1983) showed that head calculated for the block would equal the head from a single production well at a radial distance:

$$r_e = 0.28\ (\Delta x^2 + \Delta y^2)^{1/2}$$

where r_e is the equivalent well block radius.

The head in the pumping well, assuming steady flow within the well block, which is usually a good assumption, is given by:

$$h_w = h_{ijk} + Q \ln(r_e/r_w) / 2\pi T_{ijk}$$

where r_w is the well radius, and

Q is the pumping rate, $PUMP_{ijk}$, with drawal is negative.

GENERAL CONSIDERATIONS

The code can be adjusted for any size problem. Three arrays are used to store intermediate computational values in SFOR and SBAK. All the common information is passed through named COMMON statements in the program. In order to change problem dimensions of the arrays in each common block in each subroutine must be changed. In addition, the values of NX, NY and Nz, which occur in the Subroutine LOAD, must be changed. If one would like a visually meaningful result the FORMAT statements which create results contained the Subroutine LOAD, and the Subroutine OUTP, need also to be changed. For one familiar with FORTRAN these are neither difficult nor major changes. As stated above, the idea of this code was that it be sufficiently straightforward and simple so that the user could introduce significant changes with some ease.

REFERENCES

Bredehoeft, J.D. and G.F. Pinder, 1970, Digital analysis of areal flow in multiaquifer groundwater systems: a quasi three-dimensional model: Water Resources Res., v.6, p.883-999.

Hantush, M.S., 1960, Modification of the theory of leaky aquifers: Jour. Geophy. Res., v. 65, p. 3713-3725.

Peaceman, D.W., 1977, Fundamentals of numerical simulation: Elsevier, Amsterdam, 176 p.

Peaceman, D.W., 1983, Interpretation of well-block pressures in numerical reservoir simulation with non-square grid blocks and anisotropic permeability: Soc. Petrol. Eng. Jour., v. 23, p.

Peaceman, D.W. and H.H. Rachford, 1955, The numerical solution of parabolic and elliptic differential equations: SIAM Jour., v. 3, p. 28-41.

Stone, H.L., 1968, Iterative solution of implicit approximations of multidimensional partial differential equations: SIAM Jour. Numer. Anal., v. 5, p. 530-558.

Weinstein, H.E., H.L. Stone and T.V. Kwan, 1969, Iterative procedure for solution of system of parabolic and elliptic equations in three dimensions: I&EC Fundamentals, v. 8, p. 281-287.

APPENDIX I

FILE NAME	PARAMETERS TO BE LOADED	
GWPARM	S	Storage Coefficient (S = O: computes steady state)
	TOLH	Maximum Head Change for convergence
	PERD	Period of Calculations (Program multiplies PERD * 365.25 * 86400-- * seconds per year)
	TIMN	Initial Time Step
	TRMX	Multiplier for transmissivity
	CLMX	Multiplier for leakance
	FORMAT: 4F10: 4F10.5,2E10.2	
GWINTG	NITP	Number of Iteration Parameters (maximum: 20)
	NPRN	Controls Output (Output each N^{th} time step)
	IRED	Read file GWSURF as initial head--IRED = 1 to read; else read GWHEDI as initial head.
	FORMAT: 4110	
GWTHCK	THCK( )	Aquifer Saturated Thickness
	FORMAT: 20F4.1	
GWTRAN	TRAN( )	Aquifer Transmissivity
	FORMAT: 20F4.1	
GWPUMP	PUMP( )	Pumping Rate In Cell L^3/T) Withdrawal - ; Recharge +
	FORMAT: 20F4.1	
GWHEDI	HI( )	Initial Head
	FORMAT: 20F4.1	
GWSURF	HI( )	Initial Head Location (i,j,k); head
	FORMAT: 2I3,F10.4 (3D: 3I3,F10.4)	
GWWTHD	WTHD( )	Source Layer Head
	FORMAT: 20F4.1	
GWXDST	XDST( )	Array of WX Values
	FORMAT: 10F6.0	
GWYDST	YDST( )	Array of WY Values
	FORMAT: 10F6.0	
GWCLTR	CLTR( )	Leakance Values: $K_c/1_c$ (K/1) = 1: Constant Head Boundary)
	FORMAT: 20F4.1	

SIGNIFICANT PROGRAM VARIABLES

N — Number of the time step (N = 0, before calculations; N = 999 after simulation period).

NX — Number of cells in X direction.

NY — Number of cells in Y direction.

NZ — Number of layers.

ITMX — Maximum number of iterations; currently ITMX = 50--value is set in Subroutine ITER.

IN — Index of iterations--iteration number.

ITST — Condition code for convergence--ITST = 0, not converged; ITST = 1, converged.

RHO — S/Δt (Subroutine ITER).

TDEL — Δt (Subroutine ITER).

TIMN — Total elapsed time.

PARM — Iteration parameter, I.

CHCK — Absolute value of the maximum head change during any iteration.

HD(ij) — Total head change from time zero--*drawdown.*

$$HD(ij) = \sum_{N=1}^{N-1} HD^{N}(ij) + \sum_{IN=1}^{IN} [HR(ij)^{IN} + HC(ij)^{IN}]$$

HR(ij) — Head change computed in Subroutine SFOR; or in Subroutine SBAK the total change in head during the time step, exclusive of the current iteration:

$$HR(ij) = \sum_{IN=1}^{IN-1} [HC^{IN}(ij) + HR^{IN}(ij)]$$

HC(ij) — Head change computed in Subroutine SBAK; or in Subroutine SFOR the total change in head during the time step, exclusive of the current iteration:

$$HC(ij) = \sum_{IN=1}^{IN-1} [HR^{IN}(ij) + HC^{IN}(ij)]$$

AOPT(20) — Set of iteration parameters--maximum = 20.

CSTOR Cumulated volume from storge--total simulation.

$$CSTOR = \sum_{j=1}^{NY} \sum_{i=1}^{NX} S*HD(ij)\,\Delta x_i \Delta y_j$$

CPUMP Cumulative volume pumped--total simulation.

$$CPUMP = \sum_{j=1}^{NY} \sum_{i=1}^{NX} TIMN*PUMP(ij)$$

CLEAK Cumulative volume leaked in and out--total simulation.

$$CLEAK = \sum_{N=1}^{N} \sum_{j=1}^{NY} \sum_{i=1}^{NX} \frac{K_c}{l_c} [HD(ij)^N - WTHD(ij)]\,\Delta x_i \Delta y_j \Delta t^N$$

CRSTO Rate of change of storage--total area.

$$CRSTO = \sum_{j=1}^{NY} \sum_{i=1}^{NX} HC(ij)^N \frac{S}{\Delta t^N} \Delta x_i \Delta y_j$$

CRPMP Rate of pumping--total area.

$$CRPMP = \sum_{j=1}^{NY} \sum_{j=1}^{NX} PUMP(ij)$$

CLKIN Rate of leakage in--total area.

$$CLKIN = \sum_{j=1}^{NY} \sum_{i=1}^{NX} [leakage\ in(ij)]^N \Delta x_i \Delta y_j$$

CLKOT Rate of leakage out--total area.

$$CLKOT = \sum_{j=1}^{NY} \sum_{i=1}^{NX} [leakage\ out(ij)]^N \Delta x_i \Delta y_j$$

Note: Since the constant head boundaries are handled as leakance, the volume of leakages include the boundary fluxes.

RULES OF THUMB

Delta X and Delta Y

$$2 \geq \frac{\Delta x_i}{\Delta x_{i\pm1}} \geq \frac{1}{2} \; ; \; (2 \; \textit{times})$$

Transmissivity

$$10 \geq \frac{T_{ij}}{T_{i\pm1j}} \geq \frac{1}{10} \; ; \; (10 \; \textit{times})$$

Coefficients

$$10 \geq \frac{c_{ij}}{c_{i\pm1j}} \geq \frac{1}{10} \; ; \; (10 \; \textit{times})$$

TIMN

The smaller the initial time step (TIMN), the faster the solution converges, and the more accurate the solution will be.

TOLH

TOLH is the convergence criteria for the SIP iteration. The maximum head change during any iteration must be less than TOLH. The smaller TOLH the more accurate the solution will be. However small values of TOLH make more iterations necessary and slow the overall run time.

The effect of TOLH is best seen in the material balance. The material balance is a global measure of the accuracy of the solution. One wants the error in the material balance to be less than one percent. Commonly the material balance is accurate to two or three significant figures. If one finds the solution to be unacceptable for some reason, one way to improve it to lower the convergence criteria, TOLH. Lowering the value of TOLH by an order of magnitude almost always improves the material balance dramatically.

APPENDIX II: 2D Test Problem

The 2D test problem consists of a single well pumping 1.0 (L^3/T) near the southwest corner of the aquifer. The aquifer is bounded on the west, south and east sides by impermeable boundaries. To the north is a constant head boundary. The test problem is illustrated schematically in cross-section in Figure 5. The transmissivity is 0.1 (L^2/T). The initial head is 100.0 (L), with head at the constant head boundary 100.0 (L). Transient pumping is simulated for 0.1 years. During this period the flow sytem very nearly reaches steady state.

The input data is first displayed with a pause after each data set. The output is then displayed in sequence. Only the output at the end of the simulation period is called for.

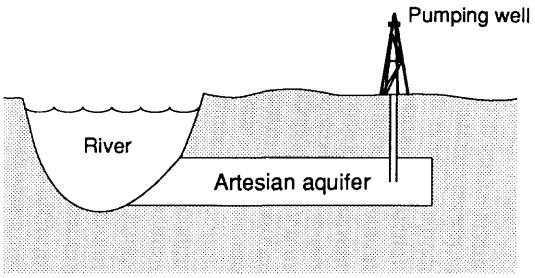

FIGURE 5. Schematic cross-section of the test problem.

PROGRAM OUTPUT

```
STORAGE COEFFICIENT = 0.100E-02
INITIAL TIME STEP   =      100.
TOLERANCE           =   0.01000
SIMULATION PERIOD   =      0.10YEARS
TRANS   MULTIPLIER  =  0.10E+00
CONFING MULTIPLIER  =  0.10E+01
ITER PARAMETERS     =  5
PRINT INDEX         = 99
READ SURF INITIAL H =  0

THICKNESS
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.

TRANSMISSIVITY
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
```

PUMPING

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0-1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

INITIAL HEAD

```
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

WATER TABLE HEAD

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	100.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

X SPACING

100.	100.	100.	100.	100.	100.	100.	100.	100.	100.
100.	100.	100.	100.	100.	100.	100.	100.	100.	100.

Y SPACING

100.	100.	100.	100.	100.	100.	100.	100.	100.	100.
100.	100.	100.	100.	100.	100.	100.	100.	100.	100.

LEAKANCE

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
TIME STEP =              0
TIME SEC  =    0.100000E+03
TIME DAYS =          0.001
TIME YEAR =          0.000


HEAD CHANGE
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.

TIME STEP =                0
CUMULATIVE STORAGE        =        0.000000E+00 L**3
CUMULATIVE PUMPING        =        0.000000E+00 L**3
CUMULATIVE LEAKAGE        =        0.000000E+00 L**3
RATE OF CHANGE OF STORAGE =        0.000000E+00 L**3/T
RATE OF PUMPING           =       -0.100000E+01 L**3/T
RATE OF LEAKAGE IN        =        0.000000E+00 L**3/T
RATE OF LEAKAGE OUT       =        0.000000E+00 L**3/T
ITERATIONS                =              0
```

```
TIME STEP =              999
TIME SEC  =    0.315576E+07
TIME DAYS =          36.525
TIME YEAR =           0.100
HEAD CHANGE
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0. -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. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2.  0.
 0. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2.  0.
 0. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -2. -2. -2.  0.
 0. -4. -4. -4. -4. -4. -4. -4. -4. -4. -3. -3. -3. -3. -3. -3. -3. -3.  0.
 0. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -3. -3. -3.  0.
 0. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -4. -4. -4. -4. -4. -4. -4.  0.
 0. -5. -5. -6. -6. -6. -6. -6. -6. -5. -5. -5. -5. -5. -4. -4. -4. -4.  0.
 0. -6. -6. -6. -6. -6. -7. -7. -6. -6. -6. -5. -5. -5. -5. -5. -5. -4.  0.
 0. -6. -7. -7. -7. -7. -7. -8. -7. -7. -6. -6. -6. -5. -5. -5. -5. -5.  0.
 0. -7. -7. -7. -7. -8. -8. -9. -8. -7. -7. -6. -6. -6. -6. -5. -5. -5.  0.
 0. -7. -7. -8. -8. -8. -9.-12. -9. -8. -7. -7. -6. -6. -6. -6. -5. -5.  0.
 0. -8. -8. -8. -8. -8. -9.-10. -9. -8. -7. -7. -7. -6. -6. -6. -6. -6.  0.
 0. -8. -8. -8. -8. -8. -9. -9. -8. -8. -8. -7. -7. -6. -6. -6. -6. -6.  0.
 0. -8. -8. -8. -8. -8. -8. -8. -8. -8. -8. -7. -7. -7. -6. -6. -6. -6.  0.
 0. -8. -8. -8. -8. -8. -8. -8. -8. -8. -8. -7. -7. -7. -6. -6. -6. -6.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
TIME STEP =            999
CUMULATIVE STORAGE         =       -0.144803E+05 L**3
CUMULATIVE PUMPING         =       -0.315576E+07 L**3
CUMULATIVE LEAKAGE         =        0.314107E+07 L**3
RATE OF CHANGE OF STORAGE  =        0.000000E+00 L**3/T
RATE OF PUMPING            =       -0.100000E+01 L**3/T
RATE OF LEAKAGE IN         =        0.100011E+01 L**3/T
RATE OF LEAKAGE OUT        =        0.000000E+00 L**3/T
ITERATIONS                 =                  1
FINAL HEAD
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.  0.
 0. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99.  0.
 0. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99. 99.  0.
 0. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98.  0.
 0. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98. 98.  0.
 0. 97. 97. 97. 97. 97. 97. 97. 97. 97. 97. 97. 97. 97. 97. 98. 98. 98.  0.
 0. 96. 96. 96. 96. 96. 96. 96. 96. 96. 97. 97. 97. 97. 97. 97. 97. 97.  0.
 0. 96. 96. 96. 96. 96. 96. 96. 96. 96. 96. 96. 96. 96. 97. 97. 97. 97.  0.
 0. 95. 95. 95. 95. 95. 95. 95. 95. 95. 95. 96. 96. 96. 96. 96. 96. 96.  0.
 0. 95. 95. 94. 94. 94. 94. 94. 94. 95. 95. 95. 95. 95. 96. 96. 96. 96.  0.
 0. 94. 94. 94. 94. 94. 93. 93. 94. 94. 94. 95. 95. 95. 95. 95. 95. 96.  0.
 0. 94. 93. 93. 93. 93. 93. 92. 93. 93. 94. 94. 94. 95. 95. 95. 95. 95.  0.
 0. 93. 93. 93. 93. 92. 92. 91. 92. 93. 93. 94. 94. 94. 94. 95. 95. 95.  0.
 0. 93. 93. 92. 92. 92. 91. 88. 91. 92. 93. 93. 94. 94. 94. 94. 95. 95.  0.
 0. 92. 92. 92. 92. 92. 91. 90. 91. 92. 93. 93. 93. 94. 94. 94. 94. 94.  0.
 0. 92. 92. 92. 92. 92. 91. 91. 92. 92. 92. 93. 93. 94. 94. 94. 94. 94.  0.
 0. 92. 92. 92. 92. 92. 92. 92. 92. 92. 92. 93. 93. 93. 94. 94. 94. 94.  0.
 0. 92. 92. 92. 92. 92. 92. 92. 92. 92. 92. 93. 93. 93. 94. 94. 94. 94.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

APPENDIX III: Listing 2D FORTRAN Code

MAIN:

```
C       JDB MODEL SIP - OCTOBER, 1989
        COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C  PUMP(20,20),XDST(20),YDST(20)
        COMMON/HEAD1/ HI(20,20),WTHD(20,20)
        COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
        COMMON/COEF1/ COEF(20,20,2),AOPT(20)
        COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
        COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
        COMMON/OUPT1/ STORE(6,100)
C       LOAD DATA
        CALL LOAD
C       CREATE COEFFICIENTS
        CALL COOF
        N=0
        CALL MATL
        CALL OUTP
C       STEP THRU TIME
        NMAX=100
        PSEC=PERD*365.25*86400.0
        IF(S.EQ.0.0) NMAX=2
        DO 10 INT=1,NMAX
        N=INT
        CALL ITER
        IF(TIMN.EQ.PSEC) GO TO 20
        IF(N.EQ.999) GO TO 20
10      CONTINUE
20      CONTINUE
        N=999
        CALL OUTP
        END
```

```
      SUBROUTINE LOAD
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
C     READ INITIAL PARAMETERS
C     PERD = PUMPING PERIOD(YEARS)
C     TIMN = INITIAL DELTA T(SEC)
C     TOLH = CONVERGENT CRITERIA - LARGEST RESIDUAL
C     TRMX = TRANSMISSIVITY MULTIPLIER
C     CLMX = CONFINING LAYER MULTIPLIER
C     NITP = NUMBER OF ITERATION PARAMETERS
C     NPRN = PRINT CONTROL - PRINT NPRN TIME STEP
C     IRED = 1 READ FILE SURF AS NEW INITIAL CONDITION
      NY=20
      NX=20
      OPEN(6,FILE='GWDATA')
      OPEN(5,FILE='GWPARM')
      READ(5,123)S,TOLH,PERD,TIMN,TRMX,CLMX
      WRITE(*,120)S
      WRITE(*,121)TIMN
      WRITE(*,122)TOLH
      WRITE(*,128)PERD
      WRITE(*,126)TRMX
      WRITE(*,127)CLMX
      WRITE(6,120)S
      WRITE(6,121)TIMN
      WRITE(6,122)TOLH
      WRITE(6,128)PERD
      WRITE(6,126)TRMX
      WRITE(6,127)CLMX
      OPEN(5,FILE='GWINTG')
      READ(5,124)NITP,NPRN,IRED
      WRITE(*,150)NITP
      WRITE(*,151)NPRN
      WRITE(*,153)IRED
      WRITE(6,150)NITP
      WRITE(6,151)NPRN
      WRITE(6,153)IRED
      PAUSE
C     LOAD THICKNESS
      OPEN(5,FILE='GWTHCK')
      READ(5,100)THCK
      WRITE(*,110)
      WRITE(*,101)THCK
      WRITE(6,110)
      WRITE(6,101)THCK
      PAUSE
C     LOAD TRANSMISSIVITY
      OPEN(5,FILE='GWTRAN')
      READ(5,100)TRAN
      WRITE(*,111)
      WRITE(6,111)
```

```
C       MULTIPLY TRAN( ) * TRMX
        DO 10 IY=1,NY
        DO 10 IX=1,NX
        TRAN(IX,IY)=TRAN(IX,IY)*TRMX
 10     CONTINUE
        WRITE(*,102)TRAN
        WRITE(6,102)TRAN
        PAUSE
C       LOAD PUMPING
        OPEN(5,FILE='GWPUMP')
        READ(5,100)PUMP
        WRITE(*,113)
        WRITE(*,100)PUMP
        WRITE(6,113)
        WRITE(6,100)PUMP
        PAUSE
C       LOAD INITIAL HEAD
        IF(IRED.EQ.1)GO TO 31
        OPEN(5,FILE='GWHEDI')
        READ(5,100)HI
        GO TO 32
C       READ SURF AS INTIAL HEAD
 31     CONTINUE
        OPEN(7,FILE='GWSURF')
        IT=NX*NY
        DO 35 ID=1,IT
        READ(7,129)IX,IY,HI(IX,IY)
 35     CONTINUE
        CLOSE(7)
 32     CONTINUE
        WRITE(*,114)
        WRITE(6,114)
        DO 30 IY=1,NY
        DO 30 IX=1,NX
        HI(IX,IY)=HI(IX,IY)*1.0
 30     CONTINUE
        WRITE(*,101)HI
        WRITE(6,101)HI
        PAUSE
C       LOAD WATER TABLE HEAD
        OPEN(5,FILE='GWWTHD')
        READ(5,100)WTHD
        WRITE(*,115)
        WRITE(6,115)
        DO 70 IY=1,NY
        DO 70 IX=1,NX
        WTHD(IX,IY)= WTHD(IX,IY)*1.0
 70     CONTINUE
        WRITE(*,101)WTHD
        WRITE(6,101)WTHD
        PAUSE
C       LOAD DELTA X
        OPEN(5,FILE='GWXDST')
```

```
      READ(5,170)XDST
      WRITE(*,116)
      WRITE(*,170)XDST
      WRITE(6,116)
      WRITE(6,170)XDST
C     LOAD DELTA Y
      OPEN(5,FILE='GWYDST')
      READ(5,170)YDST
      WRITE(*,117)
      WRITE(*,170)YDST
      WRITE(6,117)
      WRITE(6,170)YDST
      PAUSE
C     LOAD CLAY LEAKANCE(PERM/THICK)
      OPEN(5,FILE='GWCLTR')
      READ(5,100)CLTR
      WRITE(*,112)
      WRITE(*,100)CLTR
      WRITE(6,112)
      WRITE(6,100)CLTR
C     INTRODUCE FACTOR TO CREATE CONSTANT HEAD BOUNDARY
C     MULTIPLY CLTR( ) * CLMX
      DO 20 IY=1,NY
      DO 20 IX=1,NX
      IF(CLTR(IX,IY).EQ.1.0)GO TO 21
      CLTR(IX,IY)=CLTR(IX,IY)*CLMX
      GO TO 20
 21   CONTINUE
      AREA=XDST(IX)*YDST(IY)
      CONH=10.0*TRAN(IX,IY)/AREA
      CLTR(IX,IY)=CONH
 20   CONTINUE
      PAUSE
C     COMPLETES DATA LOAD
      CLOSE(5)
      CLOSE(6)
C     ADJUST PARAMETERS FOR 0 THICK
      DO 40 IY=1,NY
      DO 40 IX=1,NX
      IF(THCK(IX,IY).NE.0.0)GO TO 40
      TRAN(IX,IY)=0.0
      CLTR(IX,IY)=0.0
      HI(IX,IY)=0.0
      WTHD(IX,IY)=0.0
 40   CONTINUE
C     SET INITIAL HEADS
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      HD(IX,IY)=0.0
      HR(IX,IY)=0.0
      HC(IX,IY)=0.0
 50   CONTINUE
      RETURN
```

```
100   FORMAT(20F4.1)
101   FORMAT(20F4.0)
102   FORMAT(20F4.3)
110   FORMAT(1H ,' THICKNESS ')
111   FORMAT(1H ,' TRANSMISSIVITY ')
112   FORMAT(1H ,' LEAKANCE ')
113   FORMAT(1H ,' PUMPING ')
114   FORMAT(1H ,' INITIAL HEAD ')
115   FORMAT(1H ,' WATER TABLE HEAD ')
116   FORMAT(1H ,' X SPACING ')
117   FORMAT(1H ,' Y SPACING ')
120   FORMAT(1H ,' STORAGE COEFFICIENT =',E10.3)
121   FORMAT(1H ,' INITIAL TIME STEP   =',F10.0)
122   FORMAT(1H ,' TOLERANCE           =',F10.5)
128   FORMAT(1H ,' SIMULATION PERIOD   =',F10.2,'YEARS')
126   FORMAT(1H ,' TRANS  MULTIPLIER   =',E10.2)
127   FORMAT(1H ,' CONFING MULTIPLIER  =',E10.2)
150   FORMAT(1H ,' ITER PARAMETERS     =',I3)
151   FORMAT(1H ,' PRINT INDEX         =',I3)
153   FORMAT(1H ,' READ SURF INITIAL H =',I3)
123   FORMAT(4F10.5,2E10.2)
124   FORMAT(4I10)
129   FORMAT(2I3,F10.4)
170   FORMAT(10F6.0)
      END
```

```
      SUBROUTINE COOF
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/COEF1/ COEF(20,20,2),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
C     COEF(IX,IY,1)= X DIRECTION(IX,IX-1)
C     COEF(IX,IY,2)= Y DIRECTION(IY,IY-1)
      DO 10 ID=1,2
      DO 10 IY=1,NY
      DO 10 IX=1,NX
      COEF(IX,IY,ID)=0.0
 10   CONTINUE
      NBX=NX-1
      NBY=NY-1
      DO 20 IY=2,NBY
      IBY=IY-1
      DO 20 IX=2,NBX
      IBX=IX-1
      AX=TRAN(IX,IY)*XDST(IBX)+TRAN(IBX,IY)*XDST(IX)
      IF(AX.EQ.0.0)GO TO 25
      COEF(IX,IY,1)=2.0*TRAN(IBX,IY)*TRAN(IX,IY)/AX
 25   CONTINUE
      AY=TRAN(IX,IY)*YDST(IBY)+TRAN(IX,IBY)*YDST(IY)
      IF(AY.EQ.0.0)GO TO 20
      COEF(IX,IY,2)=2.0*TRAN(IX,IBY)*TRAN(IX,IY)/AY
 20   CONTINUE
C     COMPUTE ITERATION PARAMETERS
      DO 40 ID=1,20
      AOPT(ID)=0.0
 40   CONTINUE
C     COMPUTE MINIMUM ITERATION PARAMETER
      HMIN=1.0
      PIE2=3.141593*3.141593/2.0
      NX2=NX*NX
      NY2=NY*NY
      ANX2=FLOAT(NX2)
      ANY2=FLOAT(NY2)
      DO 50 IY=2,NBY
      DO 50 IX=2,NBX
      IF(COEF(IX,IY,1).EQ.0.0)GO TO 50
      IF(COEF(IX,IY,2).EQ.0.0)GO TO 50
      RAT=COEF(IX,IY,2)*XDST(IX)/COEF(IX,IY,1)*YDST(IY)
      HMX=PIE2/(ANX2*(1.0+RAT))
      HMY=PIE2/(ANY2*(1.0+1.0/RAT))
      IF(HMX.LT.HMIN)HMIN=HMX
      IF(HMY.LT.HMIN)HMIN=HMY
 50   CONTINUE
C     COMPUTE SEQUENCE OF ITERATION PARAMETERS
C     NITP= NUMBER OF ITERATION PARAMETERS
      IF(NITP.EQ.1) GO TO 61
      ARG=ALOG(1.0/HMIN)
      NTP1=NITP-1
      ANP1=FLOAT(NTP1)
      ALPH=EXP(ARG/ANP1)
```

COOF-Continued

```
      AOPT(1)=HMIN
      DO 60 ID=2,NITP
      IBD=ID-1
      AOPT(ID)=AOPT(IBD)*ALPH
60    CONTINUE
61    IF(NITP.EQ.1) AOPT(1)=1.0
      WRITE(*,100)
      WRITE(*,101)AOPT
      PAUSE
      RETURN
100   FORMAT(1H0,' ITERATION PARAMETERS ')
101   FORMAT(1H ,5E12.4)
      END
```

```
      SUBROUTINE ITER
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/COEF1/ COEF(20,20,2),AOPT(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
C     CALCULATE DELTA T & TOTAL TIME
      SPRD=PERD*365.25*86400.0
      IF(N.EQ.1)GO TO 11
      TDEL=TDEL*1.5
      TEST=TIMN+TDEL
      IF(TEST.GT.SPRD)TDEL=SPRD-TIMN
      IF(TDEL.EQ.0.0)GO TO 40
      TIMN=TIMN+TDEL
      IF(N.GT.1)GO TO 10
C     FIRST TIME STEP - INITIAL VALUES
 11   CONTINUE
      TDEL=TIMN
      ITMX=50
 10   CONTINUE
      RHO=S/TDEL
C     START ITERATIONS
C     SOLUTION - SIP
      WRITE(*,100)N
      NTH=0
      DO 20 IN=1,ITMX
      NIT=NITP-NTH
      PARM=AOPT(NIT)
      IS=MOD(IN,2)
      IF(IS.EQ.0)GO TO 15
      CALL SFOR
      WRITE(*,101)IN,CHCK
      GO TO 16
 15   CONTINUE
      CALL SBAK
      WRITE(*,102)IN,CHCK
 16   CONTINUE
      IF(ITST.EQ.1)GO TO 30
      IF(N.EQ.999)GO TO 40
      NTH=NTH+1
      IF(NTH.EQ.NITP)NTH=0
 20   CONTINUE
C     MAXIMUM ITERATIONS EXCEEDED
      WRITE(*,104)N
C     UPDATE HEAD
 30   CONTINUE
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      IF(THCK(IX,IY).EQ.0.0)GO TO 50
      HD(IX,IY)=HD(IX,IY)+HC(IX,IY)+HR(IX,IY)
 50   CONTINUE
C     CALCULATE FLUXES
      CALL MATL
```

ITER-Continued

```
      DO 60 IY=1,NY
      DO 60 IX=1,NX
      IF(THCK(IX,IY).EQ.0.0)GO TO 60
      HR(IX,IY)=0.0
      HC(IX,IY)=0.0
 60   CONTINUE
      WRITE(*,103)N
C     CHECK TO PRINT
      IPRT=MOD(N,NPRN)
      IF(IPRT.EQ.0)CALL OUTP
 40   RETURN
 100  FORMAT(1H0,' TIME STEP      ',I2,' INITIATED ')
 101  FORMAT(1H ,' FORWARD  ITER  ',I2,' COMPLETE MAX ERROR =',F10.5)
 102  FORMAT(1H ,' BACKWARD ITER  ',I2,' COMPLETE MAX ERROR =',F10.5)
 103  FORMAT(1H ,' TIME STEP      ',I2,' COMPLETE  ')
 104  FORMAT(1H ,' MAXIUM ITERATIONS EXCEEDED, TIME STEP N = ',I4)
      END
```

```
      SUBROUTINE SFOR
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/COEF1/ COEF(20,20,2),AOPT(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/TEMP1/ V(20,20),E(20,20),F(20,20)
      REAL LEAK
      DO 10 IY=1,NY
      DO 10 IX=1,NX
      V(IX,IY)=0.0
      E(IX,IY)=0.0
      F(IX,IY)=0.0
10    CONTINUE
      INX=NX-1
      INY=NY-1
      DO 20 IY=2,INY
      IYB=IY-1
      IYF=IY+1
      DO 20 IX=2,INX
      IF(THCK(IX,IY).EQ.0.0)GO TO 20
      IXB=IX-1
      IXF=IX+1
      AXB1=COEF(IX,IY,1)/XDST(IX)
      AXF1=COEF(IXF,IY,1)/XDST(IX)
      AYB2=COEF(IX,IY,2)/YDST(IY)
      AYF2=COEF(IX,IYF,2)/YDST(IY)
      SAXY=AXB1+AXF1+AYB2+AYF2
      HDC=HD(IX,IY)+HC(IX,IY)+HI(IX,IY)
      AXY=SAXY*HDC
      AXB=-AXB1*(HC(IXB,IY)+HD(IXB,IY)+HI(IXB,IY))
      AXF=-AXF1*(HC(IXF,IY)+HD(IXF,IY)+HI(IXF,IY))
      AYB=-AYB2*(HC(IX,IYB)+HD(IX,IYB)+HI(IX,IYB))
      AYF=-AYF2*(HC(IX,IYF)+HD(IX,IYF)+HI(IX,IYF))
      HDIF=HI(IX,IY)-WTHD(IX,IY)
      LEAK=CLTR(IX,IY)*(HDIF+HD(IX,IY)+HC(IX,IY))
      WELL=-PUMP(IX,IY)/(XDST(IX)*YDST(IY))
      ACON=RHO*HC(IX,IY)+LEAK+WELL
      RESD=AXB+AXF+AYB+AYF+AXY+ACON
      A=-SAXY-RHO-CLTR(IX,IY)
      B=AYB2/(1.0+PARM*E(IX,IYB))
      C=AXB1/(1.0+PARM*F(IXB,IY))
      D=A+PARM*B*E(IX,IYB)+PARM*C*F(IXB,IY)
C     -B*F(IX,IYB)-C*E(IXB,IY)
      E(IX,IY)=(AXF1-PARM*B*E(IX,IYB))/D
      EAB=ABS(E(IX,IY))
      IF(EAB.LT.1.0E-20)E(IX,IY)=0.0
      F(IX,IY)=(AYF2-PARM*C*F(IXB,IY))/D
      FAB=ABS(F(IX,IY))
      IF(FAB.LT.1.0E-20)F(IX,IY)=0.0
      V(IX,IY)=(RESD-B*V(IX,IYB)-C*V(IXB,IY))/D
      VAB=ABS(V(IX,IY))
      IF(VAB.LT.1.0E-20)V(IX,IY)=0.0
```

SFOR-Continued

```
 20     CONTINUE
        IY2=NY-2
        IX2=NX-2
        DO 30 IY=1,IY2
        IR=NY-IY
        IR1=IR+1
        DO 30 IX=1,IX2
        IC=NX-IX
        IC1=IC+1
        HR(IC,IR)=V(IC,IR)-E(IC,IR)*HR(IC1,IR)-F(IC,IR)*HR(IC,IR1)
        HRAB=ABS(HR(IC,IR))
        IF(HRAB.LT.1.0E-20)HR(IC,IR)=0.0
 30     CONTINUE
C       COMPUTE NEW TOTAL DRAWDOWN AND CHECK CONVERGENCE
        ITST=1
        CHCK=0.0
        DO 40 IY=1,NY
        DO 40 IX=1,NX
        CHCK1=ABS(HR(IX,IY))
        IF(CHCK1.GT.TOLH)ITST=0
        IF(CHCK1.GT.CHCK)CHCK=CHCK1
        HR(IX,IY)=HC(IX,IY)+HR(IX,IY)
        HC(IX,IY)=0.0
 40     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE SBAK
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/COEF1/ COEF(20,20,2),AOPT(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/TEMP1/ V(20,20),E(20,20),F(20,20)
      REAL LEAK
      DO 10 IY=1,NY
      DO 10 IX=1,NX
      V(IX,IY)=0.0
      E(IX,IY)=0.0
      F(IX,IY)=0.0
10    CONTINUE
      INX=NX-1
      INY=NY-1
      IY2=NY-2
      DO 20 IB=1,IY2
      IY=NY-IB
      IYB=IY-1
      IYF=IY+1
      DO 20 IX=2,INX
      IF(THCK(IX,IY).EQ.0.0)GO TO 20
      IXB=IX-1
      IXF=IX+1
      AXB1=COEF(IX,IY,1)/XDST(IX)
      AXF1=COEF(IXF,IY,1)/XDST(IX)
      AYB2=COEF(IX,IY,2)/YDST(IY)
      AYF2=COEF(IX,IYF,2)/YDST(IY)
      SAXY=AXB1+AXF1+AYB2+AYF2
      HDR=HD(IX,IY)+HR(IX,IY)+HI(IX,IY)
      AXY=SAXY*HDR
      AXB=-AXB1*(HR(IXB,IY)+HD(IXB,IY)+HI(IXB,IY))
      AXF=-AXF1*(HR(IXF,IY)+HD(IXF,IY)+HI(IXF,IY))
      AYB=-AYB2*(HR(IX,IYB)+HD(IX,IYB)+HI(IX,IYB))
      AYF=-AYF2*(HR(IX,IYF)+HD(IX,IYF)+HI(IX,IYF))
      HDIF=HI(IX,IY)-WTHD(IX,IY)
      LEAK=CLTR(IX,IY)*(HDIF+HD(IX,IY)+HR(IX,IY))
      WELL=-PUMP(IX,IY)/(XDST(IX)*YDST(IY))
      ACON=RHO*HR(IX,IY)+LEAK+WELL
      RESD=AXB+AXF+AYB+AYF+AXY+ACON
      A=-SAXY-RHO-CLTR(IX,IY)
      B=AYB2/(1.0+PARM*E(IX,IYF))
      C=AXB1/(1.0+PARM*F(IXB,IY))
      D=A+PARM*B*E(IX,IYF)+PARM*C*F(IXB,IY)
C     -B*F(IX,IYF)-C*E(IXB,IY)
      E(IX,IY)=(AXF1-PARM*B*E(IX,IYF))/D
      EAB=ABS(E(IX,IY))
      IF(EAB.LT.1.0E-20)E(IX,IY)=0.0
      F(IX,IY)=(AYF2-PARM*C*F(IXB,IY))/D
      FAB=ABS(F(IX,IY))
      IF(FAB.LT.1.0E-20)F(IX,IY)=0.0
      V(IX,IY)=(RESD-B*V(IX,IYF)-C*V(IXB,IY))/D
```

```
      VAB=ABS(V(IX,IY))
      IF(VAB.LT.1.0E-20)V(IX,IY)=0.0
 20   CONTINUE
      IX2=NX-2
      DO 30 IY=2,INY
      IR=IY
      IR1=IR-1
      DO 30 IX=1,IX2
      IC=NX-IX
      IC1=IC+1
      HC(IC,IR)=V(IC,IR)-E(IC,IR)*HC(IC1,IR)-F(IC,IR)*HC(IC,IR1)
      HCAB=ABS(HC(IC,IR))
      IF(HCAB.LT.1.0E-20)HC(IC,IR)=0.0
 30   CONTINUE
C     COMPUTE NEW TOTAL DRAWDOWN AND CHECK CONVERGENCE
      ITST=1
      CHCK=0.0
      DO 40 IY=1,NY
      DO 40 IX=1,NX
      CHCK1=ABS(HC(IX,IY))
      IF(CHCK1.GT.TOLH)ITST=0
      IF(CHCK1.GT.CHCK)CHCK=CHCK1
      HC(IX,IY)=HC(IX,IY)+HR(IX,IY)
      HR(IX,IY)=0.0
 40   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTP
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
      COMMON/COEF1/ COEF(20,20,2),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/RATE1/ CSTOR,CPUMP,CLEAK,CRSTO,CRPMP,CLKIN,CLKOT
      IF(N.GT.0)GO TO 10
      OPEN(5,FILE='GWLEVEL')
C     INITIAL TIME = 0.0
      TIMS=0.0
      TIMD=0.0
      TIMY=0.0
      IN=0
 10   CONTINUE
      TIMS=TIMN
      TIMD=TIMN/86400.0
      TIMY=TIMD/365.25
      WRITE(*,100)N
      WRITE(*,101)TIMS
      WRITE(*,102)TIMD
      WRITE(*,103)TIMY
      PAUSE
      WRITE(*,130)
      WRITE(*,104)HD
      WRITE(*,100)N
      PAUSE
      WRITE(5,100)N
      WRITE(5,101)TIMS
      WRITE(5,102)TIMD
      WRITE(5,103)TIMY
      WRITE(5,130)
      WRITE(5,200)HD
      WRITE(5,100)N
      WRITE(*,105)CSTOR
      WRITE(*,106)CPUMP
      WRITE(*,107)CLEAK
      WRITE(*,112)CRSTO
      WRITE(*,113)CRPMP
      WRITE(*,110)CLKIN
      WRITE(*,111)CLKOT
      WRITE(*,108)IN
      PAUSE
      WRITE(5,105)CSTOR
      WRITE(5,106)CPUMP
      WRITE(5,107)CLEAK
      WRITE(5,112)CRSTO
      WRITE(5,113)CRPMP
      WRITE(5,110)CLKIN
      WRITE(5,111)CLKOT
      WRITE(5,108)IN
      IF(N.LT.999)GO TO 20
C     COMPUTE FINAL HEAD - STORE HC(IX,IY)
```

```
      DO 25 IY=1,NY
      DO 25 IX=1,NX
      HC(IX,IY)=HI(IX,IY)+HD(IX,IY)
25    CONTINUE
      WRITE(*,135)
      WRITE(*,104)HC
      WRITE(5,135)
      WRITE(5,200)HC
      CLOSE(5)
C     STORE DATA FOR SURFER
      OPEN(6,FILE='GWSURF')
      DO 60 IY=1,NY
      DO 60 IX=1,NX
      WRITE(6,128)IX,IY,HC(IX,IY)
60    CONTINUE
      CLOSE(6)
20    CONTINUE
30    RETURN
100   FORMAT(1H ,' TIME STEP = ',1I15)
101   FORMAT(1H ,' TIME SEC  = ',1E15.6)
102   FORMAT(1H ,' TIME DAYS = ',1F15.3)
103   FORMAT(1H ,' TIME YEAR = ',1F15.3)
130   FORMAT(1H ,' HEAD CHANGE ')
135   FORMAT(1H ,' FINAL HEAD ')
104   FORMAT(20F4.0)
200   FORMAT(20F4.0)
105   FORMAT(1H ,' CUMULATIVE STORAGE       = ',1E20.6,' L**3 ')
106   FORMAT(1H ,' CUMULATIVE PUMPING       = ',1E20.6,' L**3 ')
107   FORMAT(1H ,' CUMULATIVE LEAKAGE       = ',1E20.6,' L**3 ')
108   FORMAT(1H ,' ITERATIONS               = ',1I15)
110   FORMAT(1H ,' RATE OF LEAKAGE IN       = ',1E20.6,' L**3/T ')
111   FORMAT(1H ,' RATE OF LEAKAGE OUT      = ',1E20.6,' L**3/T ')
112   FORMAT(1H ,' RATE OF CHANGE OF STORAGE = ',1E20.6,' L**3/T ')
113   FORMAT(1H ,' RATE OF PUMPING          = ',1E20.6,' L**3/T ')
120   FORMAT(6F10.3)
128   FORMAT(2I3,1F10.4)
      END
```

```
      SUBROUTINE MATL
      COMMON/DATA1/ THCK(20,20),TRAN(20,20),CLTR(20,20),
     C PUMP(20,20),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20),HR(20,20),HC(20,20)
      COMMON/COEF1/ COEF(20,20,2),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/RATE1/ CSTOR,CPUMP,CLEAK,CRSTO,CRPMP,CLKIN,CLKOT
      IF(N.GT.0)GO TO 10
      CPUMP=0.0
      CLEAK=0.0
      RHO=0.0
      TDEL=0.0
10    CONTINUE
      CSTOR=0.0
      CRSTO=0.0
      CRPMP=0.0
      CLKIN=0.0
      CLKOT=0.0
      DO 20 IY=1,NY
      DO 20 IX=1,NX
      IF(THCK(IX,IY).EQ.0.0)GO TO 20
      AREA=XDST(IX)*YDST(IY)
      HDIF=HI(IX,IY)-WTHD(IX,IY)
      HLEAK=HDIF+HD(IX,IY)
      RLEAK=CLTR(IX,IY)*AREA*(-HLEAK)
      IF(RLEAK.GT.0.0)CLKIN=CLKIN+RLEAK
      IF(RLEAK.LT.0.0)CLKOT=CLKOT+RLEAK
      RSTOR=RHO*AREA*HC(IX,IY)
      CRSTO=CRSTO+RSTOR
      CRPMP=CRPMP+PUMP(IX,IY)
      CSTOR=CSTOR+S*AREA*HD(IX,IY)
20    CONTINUE
      CLEAK=CLEAK+TDEL*(CLKIN+CLKOT)
      CPUMP=CPUMP+TDEL*CRPMP
      RETURN
      END
```

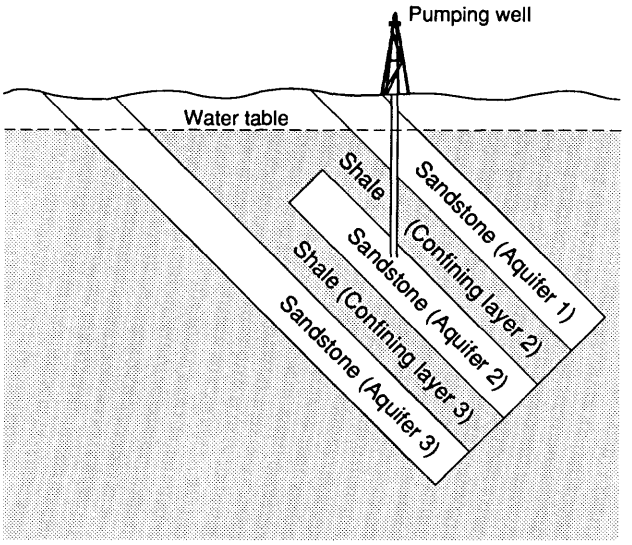

FIGURE 6. Cross-section of the test problem.

Data for a test problem is included with the 3D code. The test problem simulates three aquifers. The upper and lower aquifers outcrop along the north boundary where a constant head of 10 is maintained (the units are arbitrary). The middle aquifer does not outcrop, it is coupled to the upper and lower aquifers by an overlying and an underlying leaky confining layer. A single pumping well is pumping at three units from the middle aquifer near the southern impermeable boundary. A cross-section of the test problem is shown in Figure 6. The input is set up to simulate steady flow in this system.

Several things about the input data should be noted. Constant head is achieved by specifying the constant head in an overlying water table which is an inactive layer. At the constant head blocks a leakance value of 1.0 is specified in all three confining layers (CLTR(IX, 02,IZ)=1.0). The middle aquifer must contain a transmissivity value at the constant head blocks so the upper and lower aquifers are connected (TRAN(IX,02,02) 0.0). Otherwise the inputs are relatively straight forward; Figure 7 is a schematic cross-section of how the boundary conditions are modeled.

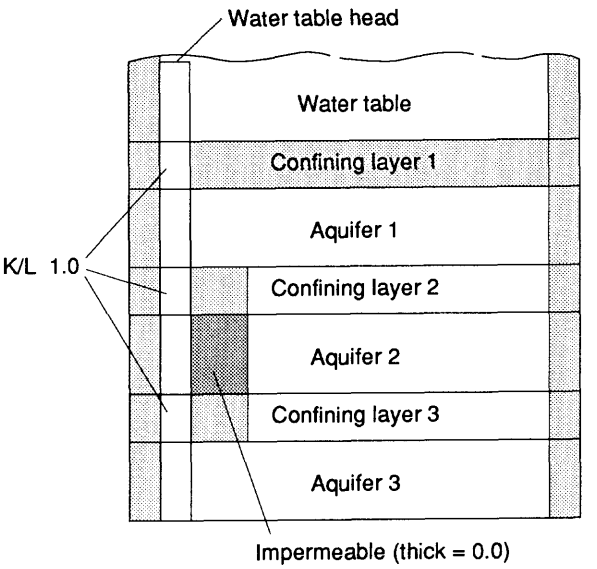

FIGURE 7. Schematic cross-section illustrating how the boundary conditions are modeled.

PROGRAM OUTPUT

```
STORAGE COEFFICIENT = 0.000E+00
INITIAL TIME STEP   =       100. T
TOLERANCE           =   0.01000
SIMULATION PERIOD   =       1.00 YEARS
TRANS MULTIPLIER    = 0.100E+00
LEAK  MULTIPLIER    = 0.100E-04
ITER PARAMETERS     =  3
PRINT INDEX         = 99
READ SURF INITIAL H =  0

THICKNESS; LAYER = 1
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  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.  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.  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.  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.  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.  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.  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.  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.  1.  0.
0.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.

THICKNESS; LAYER = 2
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  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.  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.  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.  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.  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.  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.  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.  1.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

THICKNESS; LAYER = 3

```
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  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.  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.  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.  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.  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.  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.  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.  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.  1.  0.
0.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

TRANSMISSIVITY; LAYER = 1

```
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
```

TRANSMISSIVITY; LAYER = 2

```
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
```

TRANSMISSIVITY; LAYER = 3

```
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.100.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
```

PUMPING; LAYER = 1

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

PUMPING; LAYER = 2

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

PUMPING; LAYER = 3

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

INITIAL HEAD; LAYER = 1

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

INITIAL HEAD; LAYER = 2

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

INITIAL HEAD; LAYER = 3

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

WATER TABLE HEAD

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

X SPACING

100.	100.	100.	100.	100.	100.	100.	100.	100.	100.
100.	100.	100.	100.	100.	100.	100.	100.	100.	100.

Y SPACING

100.	100.	100.	100.	100.	100.	100.	100.	100.	100.
100.	100.	100.	100.	100.	100.	100.	100.	100.	100.

LEAKANCE; LAYER = 1

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

LEAKANCE; LAYER = 2

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

LEAKANCE; LAYER = 3

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
STORAGE COEFFICIENT =  0.000E+00
TIME STEP           =  0
TIME SEC  =     0.100000E+03
TIME DAYS =            0.001
TIME YEAR =            0.000
TIME STEP           =  0

HEAD CHANGE: LAYER  =  1
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.

TIME STEP           =  0
HEAD CHANGE: LAYER  =  2
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

```
TIME STEP           =   0
HEAD CHANGE: LAYER  =   3
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
CUMULATIVE STORAGE  =          0.000000E+00 L**3
CUMULATIVE PUMP     =          0.000000E+00 L**3
CUMULATIVE LEAK     =          0.000000E+00 L**3
RATE CHANGE STORAGE =          0.000000E+00 L**3/T
RATE PUMP           =         -0.300000E+01 L**3/T
RATE LEAKAGE IN     =          0.000000E+00 L**3/T
RATE LEAKAGE OUT    =          0.000000E+00 L**3/T
TIME STEP           = 999
TIME SEC  =     0.250000E+03
TIME DAYS =            0.003
TIME YEAR =            0.000
TIME STEP           = 999
HEAD CHANGE: LAYER  =   1
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
0. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.  0.
0. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2.  0.
0. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3.  0.
0. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3.  0.
0. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4.  0.
0. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5.  0.
0. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5.  0.
0. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6.  0.
0. -6. -6. -6. -6. -6. -6. -6. -7. -7. -7. -7. -7. -7. -6. -6. -6. -6. -6.  0.
0. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7.  0.
0. -7. -7. -7. -7. -7. -7. -8. -8. -8. -8. -8. -8. -8. -8. -7. -7. -7. -7.  0.
0. -7. -7. -8. -8. -8. -8. -8. -8. -9. -9. -9. -8. -8. -8. -8. -8. -8. -8.  0.
0. -8. -8. -8. -8. -8. -8. -9. -9. -9. -9. -9. -9. -9. -9. -8. -8. -8. -8.  0.
0. -8. -8. -8. -8. -9. -9. -9.-10.-10.-10.-10.-10. -9. -9. -9. -9. -8. -8.  0.
0. -8. -8. -8. -9. -9. -9.-10.-10.-10.-11.-10.-10.-10. -9. -9. -9. -9. -9.  0.
0. -8. -8. -9. -9. -9. -9.-10.-10.-11.-11.-11.-10.-10.-10. -9. -9. -9. -9.  0.
0. -8. -9. -9. -9. -9. -9.-10.-10.-11.-11.-11.-10.-10.-10. -9. -9. -9. -9.  0.
0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
```

```
TIME STEP               = 999
HEAD CHANGE: LAYER  =   2
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4.  0.
 0. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4.  0.
 0. -4. -4. -4. -4. -4. -4. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -4.  0.
 0. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5.  0.
 0. -5. -5. -5. -5. -5. -5. -5. -6. -6. -6. -6. -6. -6. -6. -5. -5. -5.  0.
 0. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6.  0.
 0. -6. -6. -6. -6. -6. -7. -7. -7. -7. -7. -7. -7. -7. -7. -6. -6. -6.  0.
 0. -7. -7. -7. -7. -7. -7. -7. -7. -8. -8. -8. -7. -7. -7. -7. -7. -7.  0.
 0. -7. -7. -7. -7. -7. -8. -8. -8. -8. -8. -8. -8. -8. -8. -7. -7. -7.  0.
 0. -7. -7. -8. -8. -8. -8. -9. -9. -9.-10. -9. -9. -9. -8. -8. -8. -8.  0.
 0. -8. -8. -8. -8. -8. -9. -9.-10.-11.-11.-11.-10. -9. -9. -9. -8. -8.  0.
 0. -8. -8. -8. -8. -9. -9.-10.-11.-13.-14.-13.-11.-10. -9. -9. -9. -9.  0.
 0. -8. -8. -9. -9. -9.-10.-11.-12.-15.-22.-15.-12.-11.-10. -9. -9. -9.  0.
 0. -8. -8. -9. -9. -9.-10.-11.-12.-14.-15.-14.-12.-11.-10.-10. -9. -9.  0.
 0. -8. -9. -9. -9. -9.-10.-11.-12.-13.-13.-13.-12.-11.-10.-10. -9. -9.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.

TIME STEP               = 999
HEAD CHANGE: LAYER  =   3
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.  0.
 0. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2. -2.  0.
 0. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3.  0.
 0. -3. -3. -3. -3. -3. -3. -3. -4. -4. -4. -4. -4. -4. -4. -4. -3. -3.  0.
 0. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4. -4.  0.
 0. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5.  0.
 0. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5.  0.
 0. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6. -6.  0.
 0. -6. -6. -6. -6. -6. -6. -6. -7. -7. -7. -7. -7. -7. -7. -6. -6. -6.  0.
 0. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7. -7.  0.
 0. -7. -7. -7. -7. -7. -7. -8. -8. -8. -8. -8. -8. -8. -8. -7. -7. -7.  0.
 0. -7. -7. -8. -8. -8. -8. -8. -8. -9. -9. -9. -8. -8. -8. -8. -8. -8.  0.
 0. -8. -8. -8. -8. -8. -8. -9. -9. -9. -9. -9. -9. -9. -9. -8. -8. -8.  0.
 0. -8. -8. -8. -8. -9. -9. -9.-10.-10.-10.-10.-10. -9. -9. -9. -9. -8.  0.
 0. -8. -8. -8. -9. -9. -9.-10.-10.-10.-11.-10.-10.-10. -9. -9. -9. -9.  0.
 0. -8. -8. -9. -9. -9. -9.-10.-10.-11.-11.-11.-10.-10.-10. -9. -9. -9.  0.
 0. -8. -9. -9. -9. -9. -9.-10.-10.-11.-11.-11.-10.-10.-10. -9. -9. -9.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.

CUMULATIVE STORAGE  =          0.000000E+00 L**3
CUMULATIVE PUMP     =         -0.750000E+03 L**3
CUMULATIVE LEAK     =         -0.746627E+03 L**3
RATE CHANGE STORAGE =          0.000000E+00 L**3/T
RATE PUMP           =         -0.300000E+01 L**3/T
RATE LEAKAGE IN     =         -0.298776E+01 L**3/T
RATE LEAKAGE OUT    =          0.000000E+00 L**3/T
ITERATIONS          =                     2
```

TIME STEP = 999

FINAL HEAD: LAYER = 1

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	0.
0.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	0.
0.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	0.
0.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	0.
0.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	0.
0.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	0.
0.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	0.
0.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	0.
0.	4.	4.	4.	4.	4.	4.	4.	3.	3.	3.	3.	3.	3.	4.	4.	4.	4.	0.
0.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	0.
0.	3.	3.	3.	3.	3.	3.	2.	2.	2.	2.	2.	2.	2.	2.	3.	3.	3.	0.
0.	3.	3.	2.	2.	2.	2.	2.	2.	1.	1.	1.	2.	2.	2.	2.	2.	2.	0.
0.	2.	2.	2.	2.	2.	2.	1.	1.	1.	1.	1.	1.	1.	1.	2.	2.	2.	0.
0.	2.	2.	2.	2.	1.	1.	1.	0.	0.	0.	0.	0.	1.	1.	1.	1.	2.	0.
0.	2.	2.	2.	1.	1.	1.	0.	0.	0.	-1.	0.	0.	0.	1.	1.	1.	1.	0.
0.	2.	2.	1.	1.	1.	1.	0.	0.	-1.	-1.	-1.	0.	0.	0.	1.	1.	1.	0.
0.	2.	1.	1.	1.	1.	1.	0.	0.	-1.	-1.	-1.	0.	0.	0.	1.	1.	1.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

TIME STEP = 999

FINAL HEAD: LAYER = 2

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	0.
0.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	0.
0.	6.	6.	6.	6.	6.	6.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	6.	0.
0.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	0.
0.	5.	5.	5.	5.	5.	5.	5.	4.	4.	4.	4.	4.	4.	4.	5.	5.	5.	0.
0.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	0.
0.	4.	4.	4.	4.	4.	3.	3.	3.	3.	3.	3.	3.	3.	3.	4.	4.	4.	0.
0.	3.	3.	3.	3.	3.	3.	3.	3.	2.	2.	2.	3.	3.	3.	3.	3.	3.	0.
0.	3.	3.	3.	3.	3.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	3.	3.	0.
0.	3.	3.	2.	2.	2.	2.	1.	1.	1.	0.	1.	1.	1.	2.	2.	2.	2.	0.
0.	2.	2.	2.	2.	2.	1.	1.	0.	-1.	-1.	-1.	0.	1.	1.	1.	2.	2.	0.
0.	2.	2.	2.	2.	1.	1.	0.	-1.	-3.	-4.	-3.	-1.	0.	1.	1.	1.	1.	2.	0.
0.	2.	2.	1.	1.	1.	0.	-1.	-2.	-5.	-12.	-5.	-2.	-1.	0.	1.	1.	1.	1.	0.
0.	2.	2.	1.	1.	1.	0.	-1.	-2.	-4.	-5.	-4.	-2.	-1.	0.	0.	1.	1.	1.	0.
0.	2.	1.	1.	1.	1.	0.	-1.	-2.	-3.	-3.	-3.	-2.	-1.	0.	0.	1.	1.	1.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

TIME STEP = 999

FINAL HEAD: LAYER = 3

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	10.	0.
0.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	9.	0.
0.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	8.	0.
0.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	7.	0.
0.	7.	7.	7.	7.	7.	7.	7.	6.	6.	6.	6.	6.	6.	6.	6.	7.	7.	0.
0.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	6.	0.
0.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	0.
0.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	5.	0.
0.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	4.	0.
0.	4.	4.	4.	4.	4.	4.	4.	3.	3.	3.	3.	3.	3.	3.	4.	4.	4.	0.
0.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	3.	0.
0.	3.	3.	3.	3.	3.	3.	2.	2.	2.	2.	2.	2.	2.	2.	3.	3.	3.	0.
0.	3.	3.	2.	2.	2.	2.	2.	2.	1.	1.	1.	2.	2.	2.	2.	2.	2.	0.
0.	2.	2.	2.	2.	2.	2.	1.	1.	1.	1.	1.	1.	1.	1.	2.	2.	2.	0.
0.	2.	2.	2.	2.	1.	1.	1.	0.	0.	0.	0.	0.	1.	1.	1.	1.	2.	0.
0.	2.	2.	2.	1.	1.	1.	0.	0.	0.	-1.	0.	0.	0.	1.	1.	1.	1.	0.
0.	2.	2.	1.	1.	1.	1.	0.	0.	-1.	-1.	-1.	0.	0.	0.	1.	1.	1.	0.
0.	2.	1.	1.	1.	1.	1.	0.	0.	-1.	-1.	-1.	0.	0.	0.	1.	1.	1.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

APPENDIX V: Listing 3D FORTRAN Code

MAIN

```
C       JDB MODEL AUGUST,1989 - SIP VERSION
        COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C  PUMP(20,20,3),XDST(20),YDST(20)
        COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
        COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
        COMMON/COEF1/ COEF(20,20,6),AOPT(20)
        COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
        COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
        COMMON/RATE1/ CSTOR,CPUMP,CLEAK,CRSTO,CRPMP,CLKIN,CLKOT
C       LOAD DATA
        CALL LOAD
C       CREATE COEFFICIENTS
        CALL COOF
        N=0
        CALL MATL
        CALL OUTP
C       STEP THRU TIME
        NMAX=100
        PSEC=PERD*365.25*86400.0
        IF(S.EQ.0.0) NMAX=2
        DO 10 INT=1,NMAX
        N=INT
        CALL ITER
        IF(TIMN.EQ.PSEC) GO TO 20
        IF(N.EQ.999) GO TO 20
 10     CONTINUE
 20     CONTINUE
        N=999
        CALL OUTP
        END
```

```
      SUBROUTINE LOAD
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
C     READ INITIAL PARAMETERS
C     PERD = PUMPING PERIOD(YEARS)
C     TIMN = INITIAL DELTA T(SEC)
C     TOLH = CONVERGENT CRITERIA - LARGEST RESIDUAL
C     TRMX = TRANS MULTIPLIER
C     CLMX = LEAKANCE MULTIPLIER
C     NPRN = PRINT CONTROL - PRINT NTH TIME STEP
C     IRED = READ SURF AS INTIAL HEAD
      NY=20
      NX=20
      NZ=3
      OPEN(6,FILE='GWDATA')
      OPEN(5,FILE='GWPARM')
      READ(5,123)S,TOLH,PERD,TIMN,TRMX,CLMX
      S=S*1.0
      WRITE(*,120)S
      WRITE(*,121)TIMN
      WRITE(*,122)TOLH
      WRITE(*,128)PERD
      WRITE(*,130)TRMX
      WRITE(*,131)CLMX
      WRITE(6,120)S
      WRITE(6,121)TIMN
      WRITE(6,122)TOLH
      WRITE(6,128)PERD
      WRITE(6,130)TRMX
      WRITE(6,131)CLMX
      OPEN(5,FILE='GWINTG')
      READ(5,124)NITP,NPRN,IRED
      WRITE(*,150)NITP
      WRITE(*,151)NPRN
      WRITE(*,152)IRED
      WRITE(6,150)NITP
      WRITE(6,151)NPRN
      WRITE(6,152)IRED
      PAUSE
C     LOAD THICKNESS
      OPEN(5,FILE='GWTHCK')
      READ(5,100)THCK
      DO 1 IZ=1,NZ
      WRITE(*,110)IZ
      WRITE(6,110)IZ
      DO 2 IY=1,NY
      WRITE(*,101)(THCK(IX,IY,IZ),IX=1,NX)
      WRITE(6,101)(THCK(IX,IY,IZ),IX=1,NX)
 2    CONTINUE
      PAUSE
```

LOAD-Continued

```
 1      CONTINUE
C       LOAD TRANSMISSIVITY
        OPEN(5,FILE='GWTRAN')
        READ(5,100)TRAN
        DO 10 IZ=1,NZ
        DO 10 IY=1,NY
        DO 10 IX=1,NX
        IF(THCK(IX,IY,IZ).EQ.0.0)TRAN(IX,IY,IZ)=0.0
        TRAN(IX,IY,IZ)=TRAN(IX,IY,IZ)*TRMX
 10     CONTINUE
        DO 11 IZ=1,NZ
        WRITE(*,111)IZ
        WRITE(6,111)IZ
        DO 12 IY=1,NY
        WRITE(*,102)(TRAN(IX,IY,IZ),IX=1,NX)
        WRITE(6,102)(TRAN(IX,IY,IZ),IX=1,NX)
 12     CONTINUE
        PAUSE
 11     CONTINUE
C       LOAD PUMPING
        OPEN(5,FILE='GWPUMP')
        READ(5,100)PUMP
        DO 13 IZ=1,NZ
        WRITE(*,113)IZ
        WRITE(6,113)IZ
        DO 14 IY=1,NY
        WRITE(*,100)(PUMP(IX,IY,IZ),IX=1,NX)
        WRITE(6,100)(PUMP(IX,IY,IZ),IX=1,NX)
 14     CONTINUE
        PAUSE
 13     CONTINUE
C       LOAD INITIAL HEAD
        IF(IRED.EQ.1)GO TO 35
        OPEN(5,FILE='GWHEDI')
        READ(5,100)HI
        GO TO 39
 35     CONTINUE
C       READ SURF AS INITIAL HEAD
        OPEN(7,FILE='GWSURF')
        IT=NZ*NY*NX
        DO 36 ID=1,IT
        READ(7,999)IX,IY,IZ,HI(IX,IY,IZ)
 36     CONTINUE
        CLOSE(7)
 39     CONTINUE
        DO 31 IZ=1,NZ
        WRITE(*,114)IZ
        WRITE(6,114)IZ
        DO 32 IY=1,NY
        WRITE(*,101)(HI(IX,IY,IZ),IX=1,NX)
        WRITE(6,101)(HI(IX,IY,IZ),IX=1,NX)
 32     CONTINUE
        PAUSE
```

LOAD-Continued

```
 31     CONTINUE
C       LOAD WATER TABLE HEAD
        OPEN(5,FILE='GWWTHD')
        READ(5,100)WTHD
        WRITE(*,115)
        WRITE(6,115)
        WRITE(*,101)WTHD
        WRITE(6,101)WTHD
        PAUSE
C       LOAD DELTA X
        OPEN(5,FILE='GWXDST')
        READ(5,170)XDST
        WRITE(*,116)
        WRITE(6,116)
        WRITE(*,170)XDST
        WRITE(6,170)XDST
C       LOAD DELTA Y
        OPEN(5,FILE='GWYDST')
        READ(5,170)YDST
        WRITE(*,117)
        WRITE(6,117)
        WRITE(*,170)YDST
        WRITE(6,170)YDST
        PAUSE
C       LOAD CLAY LEAKANCE(PERM/THICK)
        OPEN(5,FILE='GWCLTR')
        READ(5,100)CLTR
        DO 21 IZ=1,NZ
        WRITE(*,112)IZ
        WRITE(6,112)IZ
        DO 22 IY=1,NY
        WRITE(*,100)(CLTR(IX,IY,IZ),IX=1,NX)
        WRITE(6,100)(CLTR(IX,IY,IZ),IX=1,NX)
 22     CONTINUE
        PAUSE
 21     CONTINUE
C       INTRODUCE FACTOR TO CREATE CONSTANT HEAD BOUNDARY
        DO 20 IZ=1,NZ
        DO 20 IY=1,NY
        DO 20 IX=1,NX
        AREA=XDST(IX)*YDST(IY)
        CONH=10.0*TRAN(IX,IY,IZ)/AREA
        IF(CLTR(IX,IY,IZ).LT.0.99)
    C   CLTR(IX,IY,IZ)=CLTR(IX,IY,IZ)*CLMX
        IF(CLTR(IX,IY,IZ).EQ.1.0)CLTR(IX,IY,IZ)=CONH
 20     CONTINUE
C       COMPLETES DATA LOAD
        CLOSE(5)
        CLOSE(6)
C       ADJUST PARAMETERS FOR 0 THICK
        DO 40 IZ=1,NZ
        IZB=IZ+1
        DO 40 IY=1,NY
```

```
      DO 40 IX=1,NX
      IF(THCK(IX,IY,IZ).NE.0.0)GO TO 40
      TRAN(IX,IY,IZ)=0.0
      PUMP(IX,IY,IZ)=0.0
      CLTR(IX,IY,IZ)=0.0
      IF(IZ.LT.NZ)CLTR(IX,IY,IZB)=0.0
      HI(IX,IY,IZ)=0.0
 40   CONTINUE
      DO 41 IY=1,NY
      DO 41 IX=1,NX
      IF(THCK(IX,IY,1).EQ.0.0)WTHD(IX,IY)=0.0
 41   CONTINUE
C     SET INITIAL HEADS
      DO 50 IZ=1,NZ
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      HD(IX,IY,IZ)=0.0
      HR(IX,IY,IZ)=0.0
      HC(IX,IY,IZ)=0.0
 50   CONTINUE
      RETURN
 100  FORMAT(20F4.1)
 101  FORMAT(20F4.0)
 102  FORMAT(20F4.3)
 110  FORMAT(1H ,' THICKNESS; LAYER = ',I1)
 111  FORMAT(1H ,' TRANSMISSIVITY; LAYER = ',I1)
 112  FORMAT(1H ,' LEAKANCE; LAYER = ',I1)
 113  FORMAT(1H ,' PUMPING; LAYER = ',I1)
 114  FORMAT(1H ,' INITIAL HEAD; LAYER = ',I1)
 115  FORMAT(1H ,' WATER TABLE HEAD ')
 116  FORMAT(1H ,' X SPACING ')
 117  FORMAT(1H ,' Y SPACING ')
 120  FORMAT(1H ,' STORAGE COEFFICIENT =',1E10.3)
 121  FORMAT(1H ,' INITIAL TIME STEP   =',1F10.0,' T ')
 122  FORMAT(1H ,' TOLERANCE           =',1F10.5)
 125  FORMAT(1H ,' PERMEABILTY         =',1E10.3,' L*L/T ')
 128  FORMAT(1H ,' SIMULATION PERIOD   =',1F10.2,' YEARS ')
 130  FORMAT(1H ,' TRANS MULTIPLIER    =',1E10.3)
 131  FORMAT(1H ,' LEAK  MULTIPLIER    =',1E10.3)
 150  FORMAT(1H ,' ITER PARAMETERS     =',I3)
 151  FORMAT(1H ,' PRINT INDEX         =',I3)
 152  FORMAT(1H ,' READ SURF INITIAL H =',I3)
 123  FORMAT(2F10.5,2F10.2,2E10.3)
 124  FORMAT(3I10)
 170  FORMAT(10F6.0)
 999  FORMAT(3I3,F10.4)
      END
```

```
      SUBROUTINE COOF
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
C     COEF(IX,IY,1)= X DIRECTION(IX,IX-1)
C     COEF(IX,IY,2)= Y DIRECTION(IY,IY-1)
      DO 10 IZ=1,NZ
      IL2=IZ*2
      IL1=IL2-1
      DO 10 IY=1,NY
      DO 10 IX=1,NX
      COEF(IX,IY,IL1)=0.0
      COEF(IX,IY,IL2)=0.0
 10   CONTINUE
      INX=NX-1
      INY=NY-1
      DO 20 IZ=1,NZ
      IL2=IZ*2
      IL1=IL2-1
      DO 20 IY=2,INY
      IBY=IY-1
      DO 20 IX=2,INX
      IBX=IX-1
      AX=TRAN(IX,IY,IZ)*XDST(IBX)+TRAN(IBX,IY,IZ)*XDST(IX)
      IF(AX.EQ.0.0)GO TO 25
      COEF(IX,IY,IL1)=2.0*TRAN(IBX,IY,IZ)*TRAN(IX,IY,IZ)/AX
 25   CONTINUE
      AY=TRAN(IX,IY,IZ)*YDST(IBY)+TRAN(IX,IBY,IZ)*YDST(IY)
      IF(AY.EQ.0.0)GO TO 27
      COEF(IX,IY,IL2)=2.0*TRAN(IX,IBY,IZ)*TRAN(IX,IY,IZ)/AY
 27   CONTINUE
 20   CONTINUE
C     COMPUTE ITERATION PARAMETERS
      DO 40 ID=1,20
      AOPT(ID)=0.0
 40   CONTINUE
C     COMPUTE MINIMUM ITERATION PARAMETER
      HMIN=1.0
      PIE2=3.141593*3.141593/2.0
      NX2=NX*NX
      NY2=NY*NY
      NZ2=NZ*NZ
      ANX2=FLOAT(NX2)
      ANY2=FLOAT(NY2)
      ANZ2=FLOAT(NZ2)
      DO 50 IZ=1,NZ
      IL2=IZ*2
      IL1=IL2-1
      DO 50 IY=2,INY
      DO 50 IX=2,INX
      IF(COEF(IX,IY,IL1).EQ.0.0)GO TO 50
```

```
      IF(COEF(IX,IY,IL2).EQ.0.0)GO TO 50
      RATYX=COEF(IX,IY,IL2)*XDST(IX)/COEF(IX,IY,IL1)*YDST(IY)
      RATZX=CLTR(IX,IY,IZ)*XDST(IX)/COEF(IX,IY,IL1)*10.0
      RATZY=CLTR(IX,IY,IZ)*YDST(IY)/COEF(IX,IY,IL2)*10.0
      ROH1=RATYX+RATZX
      ROH2=(1.0/RATYX)+RATZY
      IF(CLTR(IX,IY,IZ).EQ.0.0)ROH3=1.0
      IF(CLTR(IX,IY,IZ).GT.0.0)ROH3=(1.0/RATZX)+(1.0/RATZY)
      HM1=PIE2/(ANY2*(1.0+ROH1))
      HM2=PIE2/(ANX2*(1.0+ROH2))
      HM3=PIE2/(ANZ2*(1.0+ROH3))
      IF(HM1.LT.HMIN)HMIN=HM1
      IF(HM2.LT.HMIN)HMIN=HM2
      IF(HM3.LT.HMIN)HMIN=HM3
50    CONTINUE
      WRITE(*,900)HMIN
C     COMPUTE SEQUENCE OF ITERATION PARAMETERS
C     NITP= NUMBER OF ITERATION PARAMETERS
      IF(NITP.EQ.1) GO TO 61
      ARG=ALOG(1.0/HMIN)
      NTP1=NITP-1
      ANP1=FLOAT(NTP1)
      ALPH=EXP(ARG/ANP1)
      AOPT(1)=HMIN
      DO 60 ID=2,NITP
      IBD=ID-1
      AOPT(ID)=AOPT(IBD)*ALPH
60    CONTINUE
61    IF(NITP.EQ.1) AOPT(1)=1.000
      WRITE(*,100)
      WRITE(*,101)AOPT
      PAUSE
      RETURN
100   FORMAT(1H0,' ITERATION PARAMETERS ')
101   FORMAT(1H ,5E12.4)
900   FORMAT(1H0,' MIN ITER PARAMETER = ',1E10.3)
      END
```

```
      SUBROUTINE ITER
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
C     CALCULATE DELTA T & TOTAL TIME
      SPRD=PERD*365.25*86400.0
      IF(N.EQ.1)GO TO 11
      TDEL=TDEL*1.5
      TEST=TIMN+TDEL
      IF(TEST.GT.SPRD)TDEL=SPRD-TIMN
      IF(TDEL.EQ.0.0)GO TO 40
      TIMN=TIMN+TDEL
      IF(N.GT.1)GO TO 10
C     FIRST TIME STEP - INITIAL VALUES
 11   CONTINUE
      TDEL=TIMN
C     ITMX= MAXIUM NUMBER OF ITERATIONS
      ITMX=50
      ITST=1
 10   CONTINUE
      RHO=S/TDEL
C     START ITERATIONS
      WRITE(*,100)N
      NTH=0
      DO 20 IN=1,ITMX
      NIT=NITP-NTH
      PARM=AOPT(NIT)
      IS=MOD(IN,2)
      IF(IS.EQ.0)GO TO 15
      CALL SFOR
      WRITE(*,101)IN,CHCK
      GO TO 16
 15   CONTINUE
      CALL SBAK
      WRITE(*,102)IN,CHCK
 16   CONTINUE
      IF(ITST.EQ.1)GO TO 30
      IF(N.EQ.999)GO TO 40
      NTH=NTH+1
      IF(NTH.EQ.NITP)NTH=0
 20   CONTINUE
C     MAXIMUM ITERATIONS EXCEEDED
      WRITE(*,104)N
C     UPDATE HEAD
 30   CONTINUE
      DO 50 IZ=1,NZ
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      IF(THCK(IX,IY,IZ).EQ.0.0)GO TO 50
      HD(IX,IY,IZ)=HD(IX,IY,IZ)+HR(IX,IY,IZ)+HC(IX,IY,IZ)
 50   CONTINUE
```

ITER-Continued

```
C       CALCULATE FLUXES
        CALL MATL
C       RESET HR( ) & HC( )
        DO 60 IZ=1,NZ
        DO 60 IY=1,NY
        DO 60 IX=1,NX
        HR(IX,IY,IZ)=0.0
        HC(IX,IY,IZ)=0.0
 60     CONTINUE
        WRITE(*,103)N
C       CHECK TO PRINT
        IPRT=MOD(N,NPRN)
        IF(IPRT.EQ.0)CALL OUTP
 40     RETURN
 100    FORMAT(1H0,' TIME STEP ',I2,' INITIATED ')
 101    FORMAT(1H ,' FORWARD  ITER ',I2,' COMPLETE MAX ERROR =',E12.5)
 102    FORMAT(1H ,' BACKWARD ITER ',I2,' COMPLETE MAX ERROR =',E12.5)
 103    FORMAT(1H ,' TIME STEP ',I2,' COMPLETE  ')
 104    FORMAT(1H ,' MAXIMUN ITERATIONS EXCEEDED; TIME STEP = ',I4)
        END
```

```
      SUBROUTINE SFOR
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/TEMP1/ V(20,20,4),E(20,20,4),F(20,20,4),G(20,20,4)
      REAL LEAK
      NZ1=NZ+1
      DO 20 IZ=1,NZ1
      DO 20 IY=1,NY
      DO 20 IX=1,NX
      V(IX,IY,IZ)=0.0
      E(IX,IY,IZ)=0.0
      F(IX,IY,IZ)=0.0
      G(IX,IY,IZ)=0.0
20    CONTINUE
      IY1=NY-1
      IX1=NX-1
      DO 10 IZ=1,NZ
      IZA=IZ-1
      IZB=IZ+1
      ILB=IZ*2
      ILA=ILB-1
      DO 10 IY=2,IY1
      IYB=IY-1
      IYF=IY+1
      DO 10 IX=2,IX1
      IF(THCK(IX,IY,IZ).EQ.0.0)GO TO 10
      IXB=IX-1
      IXF=IX+1
      DX=COEF(IX,IY,ILA)/XDST(IX)
      FX=COEF(IXF,IY,ILA)/XDST(IX)
      BY=COEF(IX,IY,ILB)/YDST(IY)
      HY=COEF(IX,IYF,ILB)/YDST(IY)
      IF(IZ.EQ.1)ZZ=0.0
      IF(IZ.GT.1)ZZ=CLTR(IX,IY,IZ)
      IF(IZ.LT.NZ)SZ=CLTR(IX,IY,IZB)
      IF(IZ.EQ.NZ)SZ=0.0
      EXYZ=-DX-FX-BY-HY-ZZ-SZ
C     FORM THE RESIDUAL (RIGHT SIDE)
      HDC=HC(IX,IY,IZ)+HD(IX,IY,IZ)+HI(IX,IY,IZ)
      AXYZ=-EXYZ*HDC
      AXB=-DX*(HC(IXB,IY,IZ)+HD(IXB,IY,IZ)+HI(IXB,IY,IZ))
      AXF=-FX*(HC(IXF,IY,IZ)+HD(IXF,IY,IZ)+HI(IXF,IY,IZ))
      AYB=-BY*(HC(IX,IYB,IZ)+HD(IX,IYB,IZ)+HI(IX,IYB,IZ))
      AYF=-HY*(HC(IX,IYF,IZ)+HD(IX,IYF,IZ)+HI(IX,IYF,IZ))
      IF(IZ.EQ.1)HZA=0.0
      IF(IZ.GT.1)HZA=HC(IX,IY,IZA)+HD(IX,IY,IZA)+HI(IX,IY,IZA)
      AZA=-ZZ*HZA
      IF(IZ.LT.NZ)HZB=HC(IX,IY,IZB)+HD(IX,IY,IZB)+HI(IX,IY,IZB)
      IF(IZ.EQ.NZ)HZB=0.0
      AZB=-SZ*HZB
```

SFOR-Continued

```
      LEAK=0.0
      IF(IZ.EQ.1)LEAK=CLTR(IX,IY,IZ)*(HDC-WTHD(IX,IY))
      WELL=-PUMP(IX,IY,IZ)/(XDST(IX)*YDST(IY))
      ACON=RHO*HC(IX,IY,IZ)+LEAK+WELL
      R=AXB+AXF+AYB+AYF+AZA+AZB+AXYZ+ACON
C     COMPUTE SIP COEFFICIENTS
      SXYZ=EXYZ-RHO
      IF(IZ.EQ.1)SXYZ=SXYZ-CLTR(IX,IY,IZ)
      AA=ZZ/(1.0+PARM*(E(IX,IY,IZ)+F(IX,IY,IZ)))
      BB=BY/(1.0+PARM*(E(IX,IYB,IZB)+G(IX,IYB,IZB)))
      CC=DX/(1.0+PARM*(F(IXB,IY,IZB)+G(IXB,IY,IZB)))
      A=AA*E(IX,IY,IZ)
      C=BB*E(IX,IYB,IZB)
      GG=CC*F(IXB,IY,IZB)
      W=CC*G(IXB,IY,IZB)
      T=AA*F(IX,IY,IZ)
      U=BB*G(IX,IYB,IZB)
      SABC=A+C+GG+W+T+U
      D=SXYZ+PARM*SABC-CC*E(IXB,IY,IZB)
     C -BB*F(IX,IYB,IZB)-AA*G(IX,IY,IZ)
      E(IX,IY,IZB)=(FX-PARM*(A+C))/D
      EAB=ABS(E(IX,IY,IZB))
      IF(EAB.LT.1.0E-20)E(IX,IY,IZB)=0.0
      F(IX,IY,IZB)=(HY-PARM*(GG+T))/D
      FAB=ABS(F(IX,IY,IZB))
      IF(FAB.LT.1.0E-20)F(IX,IY,IZB)=0.0
      G(IX,IY,IZB)=(SZ-PARM*(W+U))/D
      GAB=ABS(G(IX,IY,IZB))
      IF(GAB.LT.1.0E-20)G(IX,IY,IZB)=0.0
      V(IX,IY,IZB)=(R-AA*V(IX,IY,IZ)-BB*V(IX,IYB,IZB)
     C -CC*V(IXB,IY,IZB))/D
      VAB=ABS(V(IX,IY,IZB))
      IF(VAB.LT.1.0E-20)V(IX,IY,IZB)=0.0
 10   CONTINUE
C     BACK SUBSTITUTION
      ITST=1.0
      CHCK=0.0
      DO 30 IZZ=1,NZ
      IZ=NZ+1-IZZ
      IZB=IZ+1
      DO 30 IYY=2,IY1
      IY=NY+1-IYY
      IYF=IY+1
      DO 30 IXX=2,IX1
      IX=NX+1-IXX
      IXF=IX+1
      IF(IZ.EQ.NZ)HZB=0.0
      IF(IZ.LT.NZ)HZB=HR(IX,IY,IZB)
      HR(IX,IY,IZ)=V(IX,IY,IZB)-E(IX,IY,IZB)*HR(IXF,IY,IZ)
     C -F(IX,IY,IZB)*HR(IX,IYF,IZ)-G(IX,IY,IZB)*HZB
      ACHK=ABS(HR(IX,IY,IZ))
      IF(ACHK.LT.1.0E-20)HR(IX,IY,IZ)=0.0
      IF(ACHK.GT.CHCK)CHCK=ACHK
```

SFOR-Continued

```
      IF(CHCK.GE.TOLH)ITST=0.0
 30   CONTINUE
C     COMPUTE CHANGE OVER THE TIME STEP
      DO 50 IZ=1,NZ
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      HR(IX,IY,IZ)=HC(IX,IY,IZ)+HR(IX,IY,IZ)
      HC(IX,IY,IZ)=0.0
 50   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SBAK
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/TEMP1/ V(20,20,4),E(20,20,4),F(20,20,4),G(20,20,4)
      REAL LEAK
      NZ1=NZ+1
      DO 20 IZ=1,NZ1
      DO 20 IY=1,NY
      DO 20 IX=1,NX
      V(IX,IY,IZ)=0.0
      E(IX,IY,IZ)=0.0
      F(IX,IY,IZ)=0.0
      G(IX,IY,IZ)=0.0
20    CONTINUE
      IY1=NY-1
      IX1=NX-1
      DO 10 IZZ=1,NZ
      IZ=NZ+1-IZZ
      IZA=IZ-1
      IZB=IZ+1
      ILB=IZ*2
      ILA=ILB-1
      DO 10 IYY=2,IY1
      IY=NY+1-IYY
      IYB=IY-1
      IYF=IY+1
      DO 10 IX=2,IX1
      IF(THCK(IX,IY,IZ).EQ.0.0)GO TO 10
      IXB=IX-1
      IXF=IX+1
      DX=COEF(IX,IY,ILA)/XDST(IX)
      FX=COEF(IXF,IY,ILA)/XDST(IX)
      BY=COEF(IX,IY,ILB)/YDST(IY)
      HY=COEF(IX,IYF,ILB)/YDST(IY)
      IF(IZ.EQ.1)ZZ=0.0
      IF(IZ.GT.1)ZZ=CLTR(IX,IY,IZ)
      IF(IZ.LT.NZ)SZ=CLTR(IX,IY,IZB)
      IF(IZ.EQ.NZ)SZ=0.0
      EXYZ=-DX-FX-BY-HY-ZZ-SZ
C     FORM THE RESIDUAL (RIGHT SIDE)
      HDC=HR(IX,IY,IZ)+HD(IX,IY,IZ)+HI(IX,IY,IZ)
      AXYZ=-EXYZ*HDC
      AXB=-DX*(HR(IXB,IY,IZ)+HD(IXB,IY,IZ)+HI(IXB,IY,IZ))
      AXF=-FX*(HR(IXF,IY,IZ)+HD(IXF,IY,IZ)+HI(IXF,IY,IZ))
      AYB=-BY*(HR(IX,IYB,IZ)+HD(IX,IYB,IZ)+HI(IX,IYB,IZ))
      AYF=-HY*(HR(IX,IYF,IZ)+HD(IX,IYF,IZ)+HI(IX,IYF,IZ))
      IF(IZ.EQ.1)HZA=0.0
      IF(IZ.GT.1)HZA=HR(IX,IY,IZA)+HD(IX,IY,IZA)+HI(IX,IY,IZA)
      AZA=-ZZ*HZA
      IF(IZ.LT.NZ)HZB=HR(IX,IY,IZB)+HD(IX,IY,IZB)+HI(IX,IY,IZB)
```

```
      IF(IZ.EQ.NZ)HZB=0.0
      AZB=-SZ*HZB
      LEAK=0.0
      IF(IZ.EQ.1)LEAK=CLTR(IX,IY,IZ)*(HDC-WTHD(IX,IY))
      WELL=-PUMP(IX,IY,IZ)/(XDST(IX)*YDST(IY))
      ACON=RHO*HR(IX,IY,IZ)+LEAK+WELL
      R=AXB+AXF+AYB+AYF+AZA+AZB+AXYZ+ACON
C     COMPUTE SIP COEFFICIENTS
      SXYZ=EXYZ-RHO
      IF(IZ.EQ.1)SXYZ=SXYZ-CLTR(IX,IY,IZ)
      AA=SZ/(1.0+PARM*(E(IX,IY,IZB)+F(IX,IY,IZB)))
      BB=HY/(1.0+PARM*(E(IX,IYF,IZ)+G(IX,IYF,IZ)))
      CC=DX/(1.0+PARM*(F(IXB,IY,IZ)+G(IXB,IY,IZ)))
      A=AA*E(IX,IY,IZB)
      C=BB*E(IX,IYF,IZ)
      GG=CC*F(IXB,IY,IZ)
      W=CC*G(IXB,IY,IZ)
      T=AA*F(IX,IY,IZB)
      U=BB*G(IX,IYF,IZ)
      SABC=A+C+GG+W+T+U
      D=SXYZ+PARM*SABC-CC*E(IXB,IY,IZ)
   C  -BB*F(IX,IYF,IZ)-AA*G(IX,IY,IZB)
      E(IX,IY,IZ)=(FX-PARM*(A+C))/D
      EAB=ABS(E(IX,IY,IZ))
      IF(EAB.LT.1.0E-20)E(IX,IY,IZ)=0.0
      F(IX,IY,IZ)=(BY-PARM*(GG+T))/D
      FAB=ABS(F(IX,IY,IZ))
      IF(FAB.LT.1.0E-20)F(IX,IY,IZ)=0.0
      G(IX,IY,IZ)=(ZZ-PARM*(W+U))/D
      GAB=ABS(G(IX,IY,IZ))
      IF(GAB.LT.1.0E-20)G(IX,IY,IZ)=0.0
      V(IX,IY,IZ)=(R-AA*V(IX,IY,IZB)-BB*V(IX,IYF,IZ)
   C  -CC*V(IXB,IY,IZ))/D
      VAB=ABS(V(IX,IY,IZ))
      IF(VAB.LT.1.0E-20)V(IX,IY,IZ)=0.0
 10   CONTINUE
C     BACK SUBSTITUTION
      ITST=1.0
      CHCK=0.0
      DO 30 IZ=1,NZ
      IZA=IZ-1
      DO 30 IY=2,IY1
      IYB=IY+1
      DO 30 IXX=2,IX1
      IX=NX+1-IXX
      IXF=IX+1
      IF(IZ.EQ.1)HZA=0.0
      IF(IZ.GT.1)HZA=HC(IX,IY,IZA)
      HC(IX,IY,IZ)=V(IX,IY,IZ)-E(IX,IY,IZ)*HC(IXF,IY,IZ)
   C  -F(IX,IY,IZ)*HC(IX,IYB,IZ)-G(IX,IY,IZ)*HZA
      ACHK=ABS(HC(IX,IY,IZ))
      IF(ACHK.LT.1.0E-20)HC(IX,IY,IZ)=0.0
      IF(ACHK.GT.CHCK)CHCK=ACHK
```

SBAK-Continued

```
      IF(CHCK.GE.TOLH)ITST=0.0
 30   CONTINUE
C     COMPUTE CHANGE OVER THE TIME STEP
      DO 50 IZ=1,NZ
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      HC(IX,IY,IZ)=HC(IX,IY,IZ)+HR(IX,IY,IZ)
      HR(IX,IY,IZ)=0.0
 50   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTP
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/RATE1/ CSTOR,CPUMP,CLEAK,CRSTO,CRPMP,CLKIN,CLKOT
      IF(N.GT.0)GO TO 10
      OPEN(5,FILE='GWLEVEL')
      WRITE(*,902)S
      WRITE(5,902)S
C     INITIAL TIME = 0.0
      TIMS=0.0
      TIMD=0.0
      TIMY=0.0
      IN=0
 10   CONTINUE
      TIMS=TIMN
      TIMD=TIMN/86400.0
      TIMY=TIMD/365.25
      WRITE(*,100)N
      WRITE(*,101)TIMS
      WRITE(*,102)TIMD
      WRITE(*,103)TIMY
      PAUSE
      DO 90 IZ=1,NZ
      WRITE(*,100)N
      WRITE(*,800)IZ
      DO 91 IY=1,NY
      WRITE(*,104)(HD(IX,IY,IZ),IX=1,NX)
 91   CONTINUE
      PAUSE
 90   CONTINUE
      WRITE(5,100)N
      WRITE(5,101)TIMS
      WRITE(5,102)TIMD
      WRITE(5,103)TIMY
      DO 80 IZ=1,NZ
      WRITE(5,100)N
      WRITE(5,800)IZ
      DO 81 IY=1,NY
      WRITE(5,200)(HD(IX,IY,IZ),IX=1,NX)
 81   CONTINUE
 80   CONTINUE
      WRITE(*,105)CSTOR
      WRITE(*,106)CPUMP
      WRITE(*,107)CLEAK
      WRITE(*,112)CRSTO
      WRITE(*,113)CRPMP
      WRITE(*,110)CLKIN
      WRITE(*,111)CLKOT
      WRITE(*,108)IN
      PAUSE
```

```
      WRITE(5,105)CSTOR
      WRITE(5,106)CPUMP
      WRITE(5,107)CLEAK
      WRITE(5,112)CRSTO
      WRITE(5,113)CRPMP
      WRITE(5,110)CLKIN
      WRITE(5,111)CLKOT
      WRITE(5,108)IN
      IF(N.LT.999)GO TO 20
C     COMPUTE FINAL HEAD - STORE HC(IX,IY,IZ)
      DO 25 IZ=1,NZ
      DO 25 IY=1,NY
      DO 25 IX=1,NX
      HC(IX,IY,IZ)=HI(IX,IY,IZ)+HD(IX,IY,IZ)
25    CONTINUE
      WRITE(*,100)N
      WRITE(*,101)TIMS
      WRITE(*,102)TIMD
      WRITE(*,103)TIMY
      PAUSE
      DO 92 IZ=1,NZ
      WRITE(*,100)N
      WRITE(*,801)IZ
      DO 93 IY=1,NY
      WRITE(*,104)(HC(IX,IY,IZ),IX=1,NX)
93    CONTINUE
      PAUSE
92    CONTINUE
      DO 82 IZ=1,NZ
      WRITE(5,100)N
      WRITE(5,801)IZ
      DO 83 IY=1,NY
      WRITE(5,200)(HC(IX,IY,IZ),IX=1,NX)
83    CONTINUE
82    CONTINUE
      CLOSE(5)
C     WRITE FINAL HEAD TO FILE GWSURF
      OPEN(6,FILE='GWSURF')
      DO 50 IZ=1,NZ
      DO 50 IY=1,NY
      DO 50 IX=1,NX
      WRITE(6,999)IX,IY,IZ,HC(IX,IY,IZ)
50    CONTINUE
      CLOSE(6)
20    CONTINUE
30    RETURN
101   FORMAT(1H ,' TIME SEC  = ',1E15.6)
102   FORMAT(1H ,' TIME DAYS = ',1F15.3)
103   FORMAT(1H ,' TIME YEAR = ',1F15.3)
104   FORMAT(20F4.0)
200   FORMAT(20F4.0)
105   FORMAT(1H ,' CUMULATIVE STORAGE  = ',1E20.6,' L**3 ')
106   FORMAT(1H ,' CUMULATIVE PUMP     = ',1E20.6,' L**3 ')
```

OUTP-Continued

```
107   FORMAT(1H ,' CUMULATIVE LEAK     = ',1E20.6,' L**3 ')
108   FORMAT(1H ,' ITERATIONS          = ',1I15)
110   FORMAT(1H ,' RATE LEAKAGE IN     = ',1E20.6,' L**3/T ')
111   FORMAT(1H ,' RATE LEAKAGE OUT    = ',1E20.6,' L**3/T ')
112   FORMAT(1H ,' RATE CHANGE STORAGE = ',1E20.6,' L**3/T ')
113   FORMAT(1H ,' RATE PUMP           = ',1E20.6,' L**3/T ')
100   FORMAT(1H ,' TIME STEP           = ',I3)
800   FORMAT(1H ,' HEAD CHANGE: LAYER  = ',I3)
801   FORMAT(1H ,' FINAL HEAD:  LAYER  = ',I3)
902   FORMAT(1H ,' STORAGE COEFFICIENT = ',1E10.3)
999   FORMAT(3I3,F10.4)
      END
```

```
      SUBROUTINE MATL
      COMMON/DATA1/ THCK(20,20,3),TRAN(20,20,3),CLTR(20,20,3),
     C PUMP(20,20,3),XDST(20),YDST(20)
      COMMON/HEAD1/ HI(20,20,3),WTHD(20,20)
      COMMON/HEAD2/ HD(20,20,3),HR(20,20,3),HC(20,20,3)
      COMMON/COEF1/ COEF(20,20,6),AOPT(20)
      COMMON/INTG1/ N,NX,NY,NZ,NITP,ITMX,IN,ITST,NPRN
      COMMON/PARM1/ S,RHO,TOLH,PERD,TIMN,PARM,CHCK,TDEL
      COMMON/RATE1/ CSTOR,CPUMP,CLEAK,CRSTO,CRPMP,CLKIN,CLKOT
      IF(N.GT.0)GO TO 10
      CPUMP=0.0
      CLEAK=0.0
      RHO=0.0
      TDEL=0.0
10    CONTINUE
      CSTOR=0.0
      CRSTO=0.0
      CRPMP=0.0
      CLKIN=0.0
      CLKOT=0.0
      DO 20 IZ=1,NZ
      DO 20 IY=1,NY
      DO 20 IX=1,NX
      IF(THCK(IX,IY,IZ).EQ.0.0)GO TO 20
      AREA=XDST(IX)*YDST(IY)
      HDIF=HI(IX,IY,1)-WTHD(IX,IY)
      HLEAK=HDIF+HD(IX,IY,IZ)
      RLEAK=CLTR(IX,IY,IZ)*AREA*HLEAK
      IF(IZ.GT.1)RLEAK=0.0
      IF(RLEAK.LT.0.0)CLKIN=CLKIN+RLEAK
      IF(RLEAK.GT.0.0)CLKOT=CLKOT+RLEAK
      RSTOR=RHO*AREA*(HC(IX,IY,IZ)+HR(IX,IZ,IZ))
      CRSTO=CRSTO+RSTOR
      CRPMP=CRPMP+PUMP(IX,IY,IZ)
      CSTOR=CSTOR+S*AREA*HD(IX,IY,IZ)
20    CONTINUE
      CLEAK=CLEAK+TDEL*(CLKIN+CLKOT)
      CPUMP=CPUMP+TDEL*CRPMP
      RETURN
      END
```