DOCUMENTATION OF A STEADY-STATE SALTWATER-INTRUSION MODEL FOR THREE-DIMENSIONAL GROUND-WATER FLOW, AND USER'S GUIDE

By D.B. Sapik

U.S. GEOLOGICAL SURVEY

Open-File Report 87-526

Prepared in Cooperation with

ISLAND COUNTY BOARD OF COMMISSIONERS and

STATE OF WASHINGTON DEPARTMENT OF ECOLOGY

Tacoma, Washington
1988

DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

District Chief
U.S. Geological Survey
1201 Pacific Avenue - Suite 600
Tacoma, Washington 98402

Copies of this report
can be purchased from:

U.S. Geological Survey
Books and Open-File Reports
Box 25425, Federal Center, Bldg. 810
Denver, Colorado 80225
Telephone (303) 236-7476

CONTENTS

ILLUSTRATIONS

TABLES

CONVERSION FACTORS

For the convenience of readers who may prefer to use metric (International System) units rather than the inch-pound units used in this report, values may be converted by using the following factors:

Multiply inch-pound unit	By	To obtain metric unit
inch (in.)	25.4	millimeter (mm)
foot (ft)	0.3048	meter (m)
mile (mi)	1.609	kilometer (km)
square foot (ft^2)	0.09290	square meter (m^2)
cubic foot (ft^3)	0.02832	cubic meter (m^3)
foot per second (ft/s)	0.3048	meter per second (m/s)
cubic foot per second (ft^3/s)	0.02832	cubic meter per second (m^3/s)

Sea level: In this report "sea level" refers to the National Geodetic Vertical Datum of 1929 (NGVD of 1929)--a geodetic datum derived from a general adjustment of the first-order level nets of both the United States and Canada, formerly called "Mean Sea Level of 1929."

CONVERSION FACTORS

Inch-pound units used in this report may be converted to International System of units (SI) by using the following conversion factors:

Multiply inch-pound unit	*By*	*To obtain SI unit*
acre-foot (acre-ft)	1,233	cubic meter
acre-foot per year (acre-ft/yr)	1,233	cubic meter per annum
cubic foot per day (ft^3/d)	0.028317	cubic meter per day
foot (ft)	0.3048	meter
foot per day (ft/d)	0.3048	meter per day
foot per year (ft/yr)	0.3048	meter per annum
foot squared per day (ft^2/d)	0.0929	meter squared per day
gallon per minute (gal/min)	0.06308	liter per second
inch (in.)	2.54	centimeter
inch per hour (in/h)	2.54	centimeter per hour
inch per year (in/yr)	2.54	centimeter per annum
mile (mi)	1.609	kilometer
square foot (ft^2)	0.0929	square meter
square mile (mi^2)	2.590	square kilometer

Degree Fahrenheit (°F) may be converted to degree Celsius (°C) by using the following equation:

$$°C = 5/9\ (°F - 32).$$

Degree Celsius (°C) may be converted to degree Fahrenheit (°F) by using the following equation:

$$°F = 9/5\ (°C) + 32.$$

The following terms and abbreviations also are used in this report:

microgram per liter (µg/L)
microsiemens per centimeter at 25 degrees Celsius (µS/cm)
milligram per liter (mg/L)

Sea level: In this report "sea level" refers to the National Geodetic Vertical Datum of 1929 (NGVD of 1929)--a geodetic datum derived from a general adjustment of the first-order level nets of both the United States and Canada, formerly called "Mean Sea Level of 1929."

DOCUMENTATION OF A STEADY-STATE SALTWATER-INTRUSION MODEL FOR THREE-DIMENSIONAL GROUND-WATER FLOW, AND USER'S GUIDE

By D. B. Sapik

ABSTRACT

An existing finite-difference model that simulates three-dimensional flow of ground water was modified to simulate steady flow of freshwater in a multiple-aquifer system containing freshwater and static saltwater. The two fluids are assumed to be immiscible, with constant but different densities, and are separated by a sharp interface. The interface position computed by the model for a test problem was in good agreement with the analytic solution for this problem. The model was developed to simulate saltwater intrusion in coastal aquifers, but it could be used to simulate flow in any aquifer system that is bounded by saltwater. This report describes modifications made to the existing numerical model and the method of locating an interface, and contains a user's guide for the model.

INTRODUCTION

This report documents a finite-difference model, herein called SSIM3D (Steady-State Saltwater-Intrusion Model for 3-Dimensional Ground-Water Flow), for simulating three-dimensional steady flow of fresh ground water in a multiple-aquifer system containing freshwater and saltwater. The saltwater is assumed to be static, with a hydrostatic pressure distribution. The two fluids are assumed to be immiscible, with constant but different densities, and the fluids are assumed to be separated by a sharp interface with continuity of pressure across the interface. The model was developed to simulate saltwater intrusion in coastal aquifers, but it could be used to simulate flow in any aquifer system that is bounded by saltwater. SSIM3D was developed by modifying Trescott's (1975) model that simulates three-dimensional flow in a multiple-aquifer system containing only freshwater of constant density. Corrections to Trescott's model that were made by Trescott and Larson (1976) and Torak (1982) are included in SSIM3D. Some of the numerical techniques used to locate a sharp interface are similar to those used by Guswa and Le Blanc (1981).

The purpose of this report is to describe the flow equations solved by the SSIM3D model, the method of locating the sharp interface, the testing of the model, and the modifications to Trescott's model. Also, this report contains a user's guide that describes the preparation of data required by the model and the job control language needed to run the model on a computer. A listing of the FORTRAN code for the model is included in the appendix.

This documentation was prepared as part of a ground-water resources investigation in Island County, Washington, that was conducted in cooperation with the Island County Board of Commissioners and the State of Washington Department of Ecology.

MATHEMATICAL BASIS FOR THE MODEL

SSIM3D is a numerical model that solves the equation for steady flow of fresh ground water in a multiple-aquifer system containing freshwater and static saltwater, and locates a sharp interface between the two fluids. The ground-water flow equation solved by SSIM3D and the equation used to locate the interface are described in this section. The procedure for solving these equations is described in "Solution of the Mathematical Equations."

The equation that describes the flow of a constant-density fluid in a multilayered aquifer system was developed by Hantush (1960) and modified, as shown below, to describe transient flow in any layer of a flow system illustrated in figure 1:

$$\frac{\partial}{\partial x}\left(T_x \frac{\partial H}{\partial x}\right) + \frac{\partial}{\partial y}\left(T_y \frac{\partial H}{\partial y}\right) + (C_z \Delta_z H)_a - (C_z \Delta_z H)_b = S\frac{\partial H}{\partial t} - W, \qquad (1)$$

where

- x, y, z are directions in a rectangular coordinate system (x and y are in a horizontal plane and z is vertical) [L];
- a, b refer to adjacent layers above and below, respectively;
- T_x, T_y are hydraulic transmissivities in the x- and y-directions $[L^2/t]$;
- C_z is a discharge coefficient for vertical leakage between adjacent layers $[t^{-1}]$;
- H is hydraulic head, in terms of freshwater, referenced to sea level [L];
- $\Delta_z H$ is vertical difference in freshwater head between adjacent layers [L];
- S is storage coefficient for a confined layer or specific yield for an unconfined layer $[L^3/L^3]$;
- t is time since the start of simulation [t]; and
- W is the sum of flow rates for sources (positive) and sinks (negative) [L/t].

The term $C_z \Delta_z H$ in equation 1 represents vertical flow between adjacent layers with finite thicknesses. An expression for C_z is presented later. This vertical flow term replaces $K_z \frac{\partial H}{\partial z}$ (K_z is vertical hydraulic conductivity) used by Hantush (1960). Equation 1 is similar to equation 4 of Trescott (1975), except that Trescott's vertical flow term was $\frac{\partial}{\partial z}\left(K_z \frac{\partial H}{\partial z}\right)$, and the term $-W$ in equation 1 replaces the term bW used by Trescott.

The steady-state form of equation 1, shown below, is used in this report to describe the flow of freshwater in an aquifer system containing freshwater and saltwater:

$$\frac{\partial}{\partial x}\left(T_x \frac{\partial H}{\partial x}\right) + \frac{\partial}{\partial y}\left(T_y \frac{\partial H}{\partial y}\right) + (C_z \Delta_z H)_a - (C_z \Delta_z H)_b = - W. \quad (2)$$

To solve equation 2 numerically for head, the equation is written in finite-difference form (see eq. 7 in Trescott, 1975). The finite-difference equation is obtained by replacing the spatial derivatives with algebraic expressions that are consistent with the subdivision of the spatial domain (for the flow system) into discrete elements. The spatial domain is subdivided into cube-like cells by superimposing a rectangular mesh over the x-y plane for each layer in the flow system. Each cell has areal dimensions of Δx and Δy, and the cell (or layer) thickness is Δz. A set of finite-difference equations, one for each model cell, is solved by the model to obtain head values at the center of each cell. For the model to solve equation 2, the user must specify values for coefficients of the head terms, estimates of the heads, constant-flux sources and sinks, and boundary conditions. All specified values are assumed to be constant within each model cell, but they can vary from cell to cell.

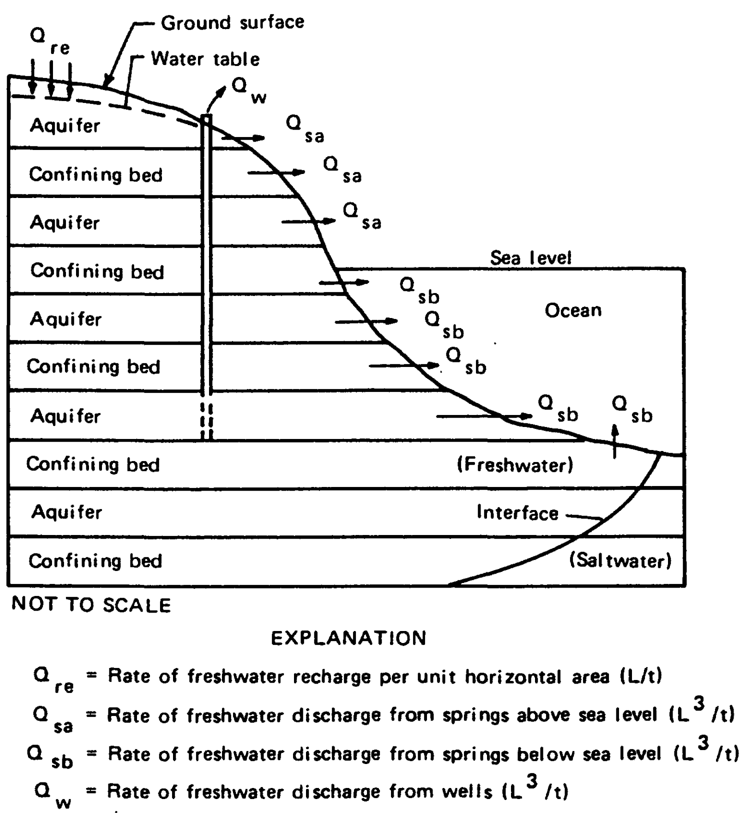


Figure 1.–Idealized schematic section for a coastal aquifer system.

Boundary conditions must be specified for each cell located along the top, bottom, and sides of the discretized flow system. The boundaries along the top and sides can be specified as constant head, constant flux, or flux that varies with head in the flow system. The flux boundary conditions are included in the source-sink term, W on the right side of equation 2, and this term also includes flux for wells that exist within the interior of the flow system. The components of the source-sink term are described below:

$$W = Q_{re} - (Q_w + Q_{sa} + Q_{sb} + Q_{rv})/(\Delta x \Delta y) \quad (3)$$

where

Q_{re} is rate of freshwater recharge per unit horizontal area [L/t]
Q_w is rate of freshwater discharge (or recharge) for wells $[L^3/t]$
Q_{sa} is rate of freshwater discharge from springs (or drains) above sea level $[L^3/t]$
Q_{sb} is rate of freshwater discharge from springs below sea level $[L^3/t]$
Q_{rv} is rate of freshwater flow (inflow is positive) between aquifers and leaky rivers $[L^3/t]$

Springs above sea level (Q_{sa}) are specified where model layers intersect land surface (shown in fig. 1) and where ground water discharges into agricultural-type drains. Springs below sea level (Q_{sb}) are specified where model layers intersect the ocean bottom (shown in fig. 1). Discharges Q_{sa}, Q_{sb}, and Q_{rv} are computed as follows:

$$\begin{aligned} Q_{sa} &= C_{sa}(H - Z_{sa}) \text{ for } H>Z_{sa} \\ &= 0 \qquad\qquad\quad \text{ for } H\le Z_{sa} \end{aligned} \quad (4)$$

and

$$\begin{aligned} Q_{sb} &= C_{sb}(H - H_{sb}) \text{ for } H>H_{sb} \\ &= 0 \qquad\qquad\quad \text{ for } H\le H_{sb} \end{aligned} \quad (5)$$

and

$$\begin{aligned} Q_{rv} &= C_{rv}(H - H_{rv}) \text{ for } H>Z_{rv} \\ &= C_{rv}(Z_{rv} - H_{rv}) \text{ for } H\le Z_{rv} \end{aligned} \quad (6)$$

where

C_{sa}, C_{sb}, and C_{rv} are discharge coefficients $[L^2/t]$;

H is hydraulic head in the freshwater zone computed from equation 2 [L];

H_{sb} is hydraulic head in the saltwater zone, in terms of freshwater, at the top of a cell [L];

H_{rv} is water-surface elevation in a river, referenced to sea level [L];

Z_{sa} is elevation at the bottom of a cell or a drain [L]; and

Z_{rv} is elevation at the bottom of a leaky riverbed [L].

The model computes vertical leakage between layers using the terms $C_z \Delta_z H$ in equations 1 and 2; discharges from springs and rivers are computed using equations 4 to 6. Values of the discharge coefficients C_z, C_{sa}, C_{sb}, and C_{rv} used in equations 1 to 2 and 4 to 6 must be specified by the model user. The computational formulas for these coefficients may not be the same for each flow system that is simulated, but as a starting point, the formulas shown below have been derived using Darcy's law. The values of these coefficients are usually adjusted during model calibration. The discharge coefficient for vertical leakage between a layer and an overlying layer (eq. 2) is

$$C_z = \frac{(2)(K_z)(K_z^a)}{(K_z)(\Delta z^a) + (K_z^a)(\Delta a)} \quad ; \qquad (7)$$

the coefficient used to compute discharge from springs above sea level (eq. 4) is

$$C_{sa} = (2)(T_x)(\Delta y)/(\Delta x) \quad \text{for discharge in the x-direction,} \qquad (8)$$

$$= (2)(T_y)(\Delta x)/(\Delta y) \quad \text{for discharge in the y-direction;}$$

the coefficient used to compute discharge from springs below sea level (eq. 5) is

$$C_{sb} = (2)(K_z)(\Delta x)(\Delta y)/(\Delta z) \quad \text{for discharge in the z-direction,} \qquad (9)$$

$$= (2)(T_x)(\Delta y)/(\Delta x) \quad \text{for discharge in the x-direction,}$$

$$= (2)(T_y)(\Delta x)/(\Delta y) \quad \text{for discharge in the y-direction;}$$

and the coefficient used to compute discharge between aquifers and leaky rivers (eq. 6) is

$$C_{rv} = K_{rv} A_{rv}/M_{rv} \quad ; \qquad (10)$$

where

- a refers to an adjacent layer above
- K_z is vertical hydraulic conductivity for a model cell [L/t]
- K_{rv} is hydraulic conductivity for a riverbed within a model cell [L/t]
- A_{rv} is area of a river bed within a model cell [L^2]
- M_{rv} is thickness of a riverbed within a model cell [L]

The model will simulate transient or steady flow of freshwater in a multiple-aquifer system containing only freshwater using equations 1 or 2 with discharges and discharge coefficients defined by equations 3 to 10.

For simulating the movement of freshwater in a ground-water flow system containing freshwater and saltwater, a sharp interface is assumed to separate moving freshwater and static saltwater. These two fluids are assumed to be immiscible, with constant but different densities. Fluid pressure in the saltwater zone is assumed to vary hydrostatically and pressure is assumed to be continuous across the interface. Along the sharp interface, pressures in the freshwater and saltwater zones are equal, and the depth to the interface is determined from an equation derived by Hubbert (1940) and modified, as shown below, for a flow system in which saltwater is static and saltwater head is at sea level:

$$Z^i = \frac{\rho_f}{\rho_s - \rho_f} H^i \qquad (11)$$

where

- Z^i = depth below sea level to the freshwater-saltwater interface [L]
- H^i = hydraulic head (computed from eq. 2) in the freshwater zone at the interface, referenced to sea level [L]
- ρ_s = density of saltwater [M/L^3]
- ρ_f = density of freshwater [M/L^3]

To simulate the steady flow of freshwater in a multiple-aquifer system containing freshwater and saltwater, the model solves equation 2 for freshwater head with discharges and discharge coefficients defined by equations 3 to 10, and uses equation 11 to locate a sharp interface between the two fluids. The procedure for solving these equations is described in "Solution of the Mathematical Equations."

SOLUTION OF THE MATHEMATICAL EQUATIONS

SSIM3D solves equations 2 and 11 iteratively using the four tasks shown in figure 2. The flow equations are solved for head in each model cell (task 2 in fig. 2) by using the strongly implicit procedure (SIP) described by Trescott (1975). For each saltwater iteration shown in figure 2, new interface positions are computed in task 4 and the transmissivities of all partly intruded cells are recomputed to account for the changed freshwater thicknesses in the cells. Spring discharge coefficients and well discharges are changed proportionately, the same as transmissivities, for partly intruded cells. Saltwater iterations are repeated until differences between damped interface positions (described later) and undamped interface positions (from eq. 11) are less than some user-specified value. Some of the numerical techniques used in subroutines FWSW and ZFWSW (task 4 in fig. 2) are similar to those used by Guswa and Le Blanc (1981); however, the subroutines developed by Guswa and Le Blanc were not used because the interface-location procedure was found to be unstable.

Saltwater intrusion occurs in the model where the computed interface position is above the bottom of a cell, as shown in figure 3. The interface is assumed to be horizontal within an intruded cell, but the vertical position of the interface can vary laterally between cells to simulate a sloping interface.

To stabilize the iterative procedure for computing the interface position, movement of the interface between successive saltwater iterations is damped, and for intruded cells, both spring discharge coefficients (see eq. 5) and well discharges (of freshwater) are changed proportionately the same as transmissivities. Because well discharges are reduced for partly intruded cells and are set to zero for totally intruded cells, total freshwater pumpage for a model simulation will differ from that specified in the model input. The consequences of reduced freshwater pumpage should be considered by a user when a model is being calibrated.

Methods of damping interface movement between successive saltwater iterations for different cases of intrusion are shown in figure 4 and are also described in the following paragraphs. No damping is necessary if a cell remains totally intruded or unintruded (contains no saltwater) in successive iterations (cases I and II in fig. 4).

If the interface is below the bottom of a cell, the maximum upward movement allowed in the following iteration is to the bottom of the cell (case III in fig. 4). If the interface is within or at the bottom of a cell, movement of the interface between two successive iterations is a user-specified fraction of the lesser of the distance between the old damped and new undamped positions, or the distance between the old damped position and the top or bottom of the cell (case IV, step 1 in fig. 4). A partly intruded cell is made totally intruded when the damped unintruded thickness of the cell becomes less than a user-specified amount; and the cell is made totally unintruded when the damped unintruded thickness becomes greater than a user-specified amount (case IV, step 2 in fig. 4).

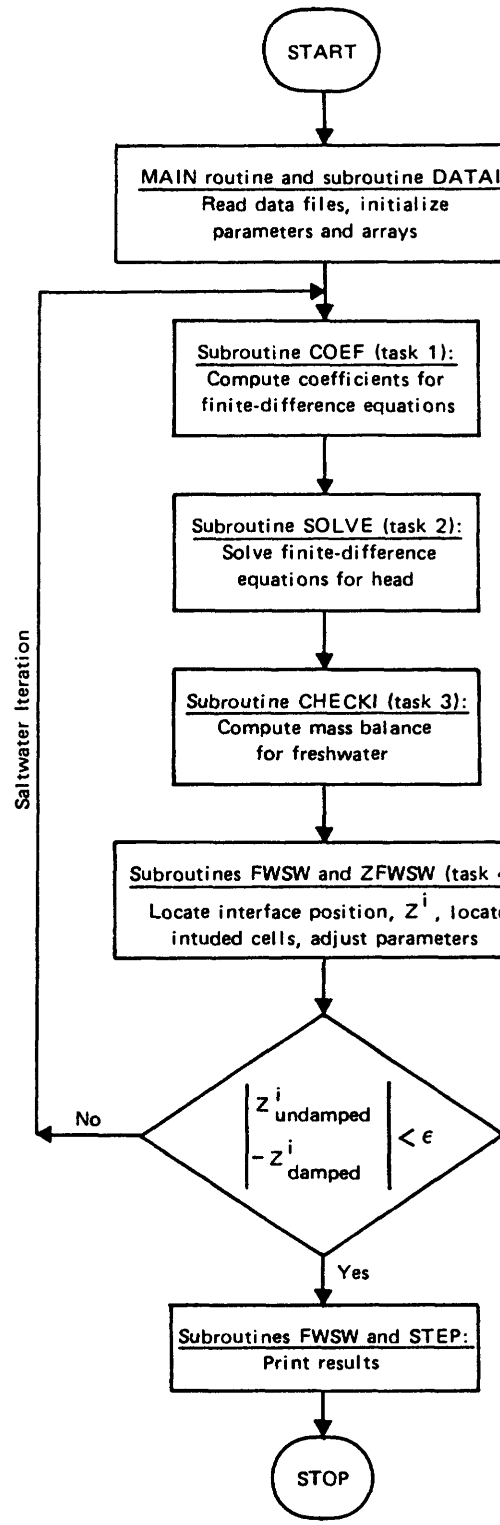


Figure 2.--Generalized flow chart for tasks performed in the saltwater-intrusion model.

If a cell is totally intruded, it is made partly intruded if the head in any adjacent cell is sufficiently large to move the interface below the top of the cell. In this case, the interface is put at a user-specified fraction of the total cell thickness below the top of the cell (case V in fig. 4). However, if a cell in the water-table layer of the model is totally intruded, it will remain so in later saltwater iterations.

To prevent small groups of cells from becoming isolated (totally surrounded by saltwater), a cell is made totally intruded if more than a user-specified number of adjacent cells are totally intruded.

The types of sharp interface problems that can be solved by SSIM3D are limited to those for which (1) there is no flow across the interface (immiscible fluids), (2) the two fluids have constant but different densities, and (3) the saltwater is static. These assumptions are the basis of computations for vertical leakage of freshwater between layers and for freshwater discharge from springs below sea level. The model does not compute vertical leakage of freshwater into cells that are totally intruded because this would violate the assumption of no flow across the interface; therefore, the model cannot be used to solve these types of problems that are described by Mualem and Bear (1974) and Bear (1979, p. 396-399). The model does compute vertical leakage of freshwater into cells that are partly intruded using the terms $C_z \Delta_z H$ in equation 2, but saltwater in the intruded cells is assumed to be static and the fluids are assumed to be immiscible. The computation of freshwater discharge from partly intruded spring cells below sea level is made by using equation 5 with the assumptions of static saltwater and immiscible fluids.

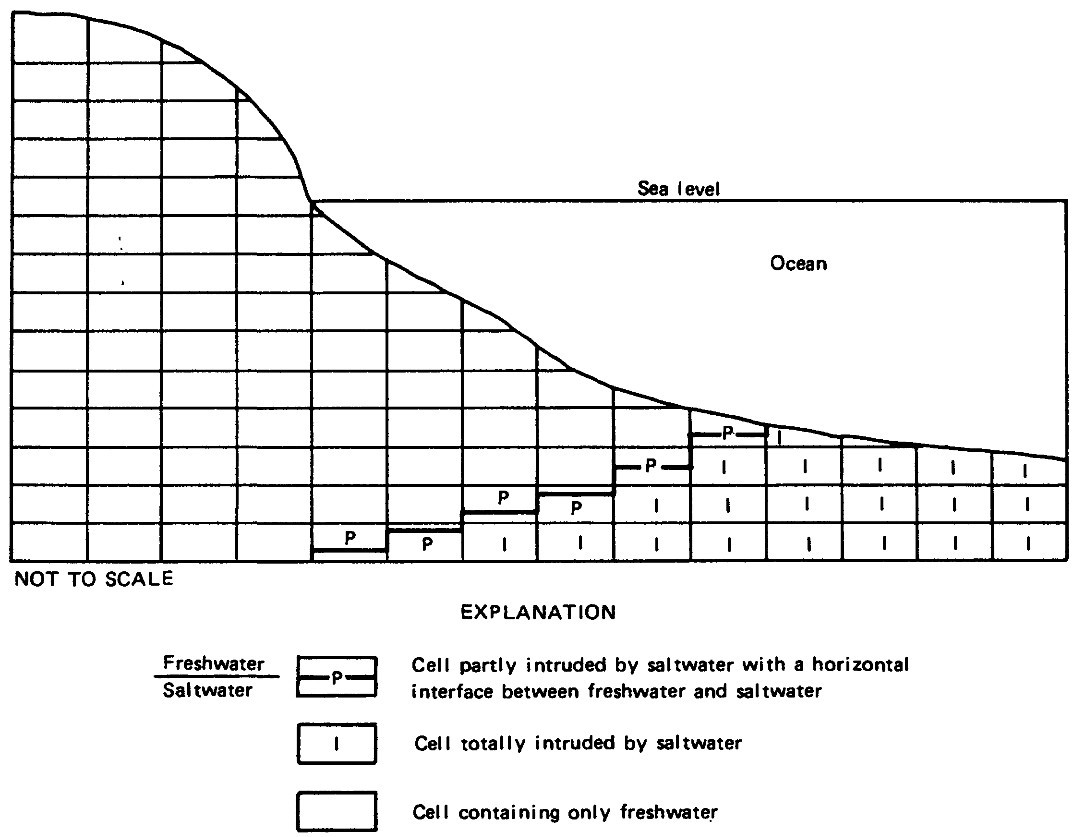
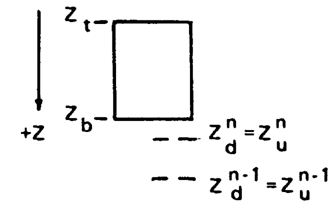
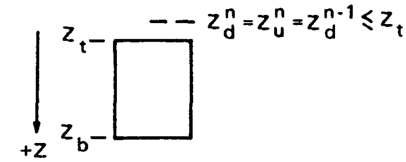


Figure 3.–Schematic section showing how the model approximates a sharp interface between freshwater and saltwater.

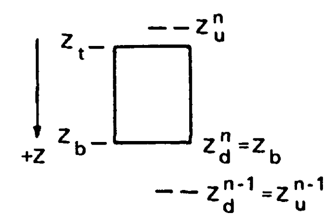
Case I: Cell is not intruded by saltwater in the previous and current saltwater iterations



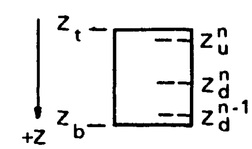
Case II: Cell is totally intruded by saltwater in the previous and current saltwater iterations



Case III: Cell is not intruded by saltwater in the previous saltwater iteration



Case IV: Cell is partly intruded by saltwater in the previous saltwater iteration



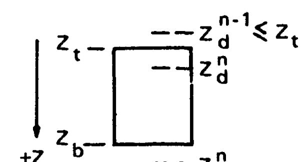
Step 1: $Z_d^n = Z_d^{n-1} - \mathrm{Min}\left[(Z_d^{n-1} - Z_t),\ (Z_d^{n-1} - Z_u^n)\right] * D_f$, or

$Z_d^n = Z_d^{n-1} + \mathrm{Min}\left[(Z_b - Z_d^{n-1}),\ (Z_u^n - Z_d^{n-1})\right] * D_f$

Step 2: $Z_d^n = Z_t$ if $(Z_d^n - Z_t) < T_{mi}$, or

$Z_d^n = Z_b$ if $(Z_d^n - Z_t) > T_{mu}$

Case V: Cell is restored to partial intrusion in the current saltwater iteration



$Z_d^n = Z_t + T_{mr}$

Figure 4.--Methods of moving the interface for cases I to V.
(Symbols are defined in explanation)

EXPLANATION

Symbol

Z_t = Depth below sea level to the top of a cell or to the water table.

Z_b = Depth below sea level to the bottom of a cell.

Z_u^{n-1} = Depth below sea level to the undamped interface position in the previous saltwater iteration (computed from eq. 11).

Z_u^n = Depth below sea level to the undamped interface position in the current saltwater iteration (computed from eq. 11).

Z_d^{n-1} = Depth below sea level to the damped interface position in the previous saltwater iteration.

Z_d^n = Depth below sea level to the damped interface position in the current saltwater iteration.

D_f = Damping factor that is used to limit the distance that the interface can move in a saltwater iteration.

T_{mi} = Minimum freshwater thickness that is used when a partly intruded cell is made totally intruded.

T_{mu} = Maximum freshwater thickness that is used when a partly intruded cell is made totally unintruded.

T_{mr} = Minimum freshwater thickness that is used when a totally intruded cell is restored to partial intrusion.

— — = Interface position

Figure 4.–Con.

MODIFICATIONS TO THE TRESCOTT MODEL

Most of the subroutines in Trescott's model (1975) were modified; however, the computational procedure for solving the ground-water flow equation (eq. 1 or 2) was not changed. The modifications consisted of correcting errors described by Trescott and Larson (1976) and by Torak (1982), reformatting some of the input and output (including printout) files, adding subroutines FWSW and ZFWSW to compute the position of a sharp interface between freshwater and saltwater, adding new FORTRAN code to the main routine and to subroutines CHECKI, COEF, DATAI, SOLVE, and STEP to support the new interface-location routines, and adding subroutines ERMES1, GWDBAS, MSQDIF, and VSEC. The corrections suggested by Torak (1982) for simulation of ground-water flow in nonlayered systems (eq. 3 of Trescott, 1975) were not made. A listing of the FORTRAN code for the new computer model is contained in Appendix I, and all changes are indicated by asterisks (*) in column 80 of the listing. A summary of the changes is presented below.

1. Global changes.
   a. REAL*8 statements: Added TFLX, CFLX, CTFLX, and SUMFLX for use in subroutine CHECKI.
   b. COMMON/INTEGR/ statements:
      (1) Deleted I, J, K because these are index variables for DO loops, values of these variables are redefined in each subroutine, and leaving these variables in COMMON downgrades optimization by the FORTRAN compiler.
      (2) Added new variables after IEQN to support additions to the model.
   c. COMMON/SPARAM/ statements:
      (1) Deleted QR because the value was not used in called subroutines.
      (2) Added new variables after SUMP to support additions to the model.
   d. COMMON/SARRAY/ statements:
      (1) Changed ICHK(13) to ICHK(20) so more options could be used.
      (2) Changed LEVEL1(9) to LEVEL1(40) and LEVEL2(9) to LEVEL2(40) so that a maximum of 40 model layers can be specified.
      (3) Added NODCH(3,51) for use in subroutine SOLVE.
   e. COMMON/CK/ statements: Replaced all variables with new arrays used in subroutine CHECKI.
   f. All subroutines:
      (1) Multiple RETURN and STOP statements in subroutines were replaced with GO TO statements and only one RETURN and one STOP statement at the end of each routine. These changes were made to facilitate tracing program logic with a cross-reference map.
      (2) Added more comments to make the program logic easier to follow.
      (3) Most of the printout in Trescott's model was reformatted to identify the data group for data read into the model, and printout was left-justified for easier viewing on a video-display terminal. All of the parameter values read into the model are printed, and messages are printed when an array data set is read into the model.

2. MAIN routine:
   a. Added IYSIZE, MXLAY, and MXITR to determine if user specifications exceed sizes of dimensioned variables.
   b. New variables were added to READ statements and new READ statements were added to support additions to the model.
   c. Added statements to allocate space for arrays used to compute discharge from drains (or springs above sea level) and leaky rivers (or springs below sea level), and arrays used in subroutine FWSW.
   d. Calls to entry points ARRAY and MDAT were deleted when subroutine DATAI was rewritten.
   e. Calls were added to new subroutines and entry points: VSEC, CHECKR, FWSW, and MSQDIF.
   f. New parameters were added to call lists for subroutines DATAI, STEP, SOLVE, and CHECKI to support additions to the model.
   g. Between statement numbers 150 and 158, new statements were added for the iterative procedure used to locate the interface.
3. BLOCK DATA routine:
   a. Added EQN3 to the DATA statement for ICHK.
   b. Made corrections to the DATA statement for DIGIT.
4. Subroutine CHECKI:
   a. Rewritten to compute and print a mass balance for each model layer. Double-precision arithmetic was used to reduce the loss of significant numbers for mass balance computations and the arrays FLX, TFLX, CFLX, CTFLX, and SUMFLX were added to COMMON/CK/.
   b. Added new code to compute discharge into drain and leaky-river cells (S.P. Larson, U.S. Geological Survey, written commun., 1976).
   c. Added new code to print or save on disk file the flux values for each constant-head, drain (or spring above sea level), and leaky-river (or spring below sea level) cell. The variables FLOW and JFLO, previously used to save constant-head flux, are no longer used but were left in the program.
   d. Code added by Trescott and Larson (1976) was rewritten to eliminate the need for redundant code when the EQN3 option is specified.
   e. Added entry point CHECKR to read coordinates of subareas for which mass balance is computed.
   f. Added new code to print vertical flux between all model layers and deleted old code that printed vertical flux only for top and bottom model layers.
5. Subroutine COEF.
   a. Modified to compute transmissivity and harmonic means for a model with water-table aquifer in layer number LWTABL.
   b. Modified to compute harmonic coefficients more efficiently (reduce computation time), because this routine is called before each SIP iteration.
   c. When a water-table cell goes dry, set PHI(I,J,K)=-99 instead of 1.D30 because -99 is used to designate cells in which transmissivity is zero.
   d. Deleted arguments in call lists for entry points TRANS and TCOF.
   e. Replaced the variable N3 with NTR (in COMMON/INTEGR/) to indicate which layers are included in the computation of harmonic means. For the saltwater intrusion problem, NTR is initialized in the MAIN routine so that confined layers are always included in the computations.
   f. The array FACT was changed to an array with three subscripts (see comments for subroutine DATAI).

6. Subroutine DATAI.
   a. Added new code to read data for simulating discharge into drains and leaky rivers (S.P. Larson, U.S. Geological Survey, written commun., 1976), to determine the uppermost layers to be recharged, and to read data for simulating saltwater intrusion.
   b. All array data sets are read by calling subroutine GWDBAS (replaces entry points ARRAY and MDAT).
   c. Read values of LEVEL1 and LEVEL2 only when YSCALE is negative.
   d. Initialize cumulative mass balance parameter, CFLX, for each model layer. A new input format is used for reading cumulative mass balance parameters for each layer.
   e. The array FACT was changed to an array with three subscripts so that data for this array can be read with subroutine GWDBAS.
   f. The model layers that are recharged can be read with subroutine GWDBAS, or a search procedure can be used to determine the topmost active layers in the model.
   g. A new input format is used for reading head values to continue a previous simulation.
   h. Set vertical leakage coefficients between layers to zero when transmissivity is zero in either of two contiguous layers.
   i. Print error messages and stop program execution if the number of constant-head cells exceeds NCH, or the number of drain cells exceeds NDRAIN, or the number of leaky-river cells exceeds NRIV.
   j. Added options to read new recharge rates, new heads for leaky-river cells, or read new heads for constant-head cells at the start of a new stress period.
7. Subroutine ERMES1.
   a. New subroutine that prints row, column, and layer coordinates for a cell when floating-point underflow, overflow, or divide check occur in subroutine SOLVE.
   b. This routine is called from subroutine ERRSET (part of the IBM FORTRAN library).
8. Subroutine FWSW.
   a. New subroutine used to locate the position of a sharp interface between freshwater and saltwater.
   b. This routine is called from the MAIN routine when ISALT=1 and is called after new head values are computed in subroutine SOLVE.
9. Subroutine GWDBAS.
   a. New subroutine used to read data for model arrays.
   b. This routine is called from subroutines DATAI and FWSW.
10. Subroutine MSQDIF.
    a. New subroutine used to compute root-mean square for differences between observed and model-computed head.
    b. This routine is called from the MAIN routine when MSQCAL=1.
11. Subroutine PRNTAI.
    a. Changed the print file to FORTRAN file number 9.
12. Subroutine SOLVE.
    a. Added new code to simulate discharge into drains and leaky rivers (S.P. Larson, U.S. Geological Survey, written commun., 1976).
    b. Added a call to IBM library subroutine ERRSET that calls subroutine ERMES1 when the results of floating-point arithmetic are underflow, overflow, or divide check.

c. Print error message and set transmissivity to zero when value of DL is less than 1.E-20.
d. Added the array NODCH to save coordinates of cells with maximum head change in each iteration.
e. Added a call to entry point TCOF that recomputes transmissivity coefficients for confined layers when transmissivity is changed because of saltwater intrusion.
f. Set coefficients of vertical leakage between layers to zero if transmissivity is zero in an overlying or underlying layer (changed in entry points ITER and NEWITA).
g. Added code to permit the user to specify the value of the last iteration parameter for SIP.
h. Added code to permit the use of an acceleration (or damping) parameter for SIP.
i. Added MXRHOP to check if the number of iteration parameters exceeds the size of the array RHOP.

13. Subroutine STEP.
    a. Revised code for printing head and drawdown maps.
    b. Added code to write single-precision head values on disk file. Before writing this file, head values are set to -99 where transmissivity is zero.
14. Subroutine VSEC.
    a. New subroutine used to print or save on disk the values of head and saltwater-intrusion codes along selected vertical sections in the model.
    b. This routine is called from the MAIN routine when NVSEC is greater than zero.
15. Subroutine ZFWSW.
    a. New subroutine used to compute the damped position of the interface.
    b. This routine is called from subroutine FWSW.

TESTING THE MODEL

SSIM3D was tested to verify that (1) changes in the FORTRAN code do not affect the results of simulating three-dimensional flow of a constant-density fluid, and (2) there is good agreement between the interface position computed from a known analytic solution to a problem and the model-computed interface for the same problem. To verify that changes in the FORTRAN code do not affect the results of simulation, an example problem solved by Trescott's (1975) model was solved by SSIM3D and the results were identical.

To test the saltwater-intrusion routines in SSIM3D, a steady-state problem solved by Glover (1959) was solved by the model. For this two-dimensional problem, Glover assumed (1) the aquifer is confined along the top by an impermeable bed, (2) the aquifer extends infinitely in the x- and z-directions, (3) the aquifer is homogeneous and isotropic, (4) the aquifer discharges freshwater through a horizontal outflow face, and (5) saltwater in the aquifer is static. A schematic for this problem is shown in figure 5. The equation derived by Glover (1959) that describes the steady-state position of the interface is

$$Z^{i} = \left[\left(\frac{2Q\,\rho_f}{K\,\Delta\rho}\right) X + \left(\frac{Q\,\rho_f}{K\,\Delta\rho}\right)^{2}\right]^{1/2} \qquad (12)$$

where

Z^i = depth below sea level to the freshwater-saltwater interface [L]

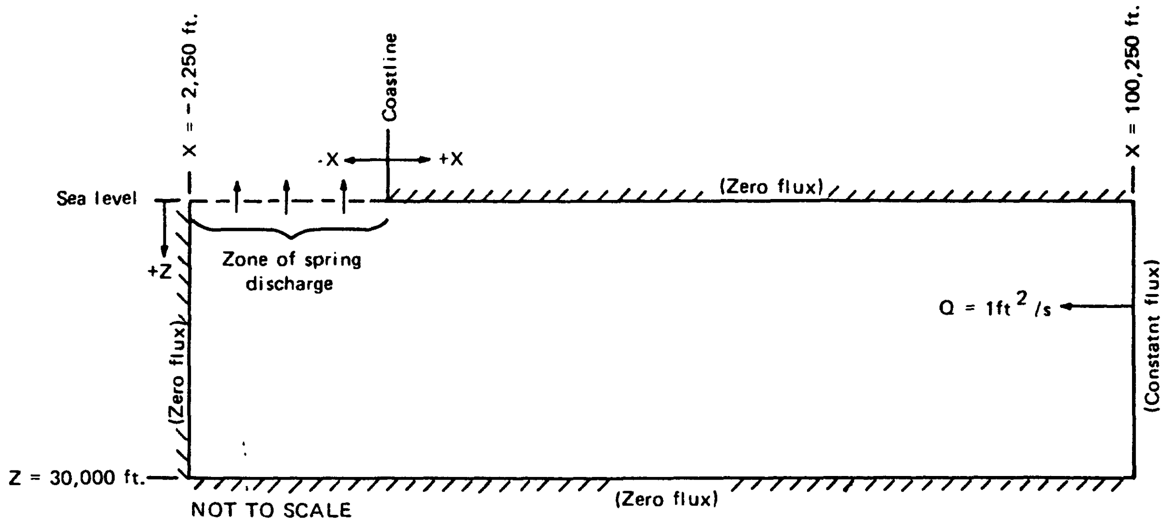
Q = freshwater-discharge rate per unit width of aquifer $[L^2/t]$

K = hydraulic conductivity of aquifer material [L/t]

X = distance from the coastline [L]

$\Delta\rho = \rho_s - \rho_f$ (see eq. 11) $[M/L^3]$

Differences in the interface position computed by the model and by equation 12 are less than 10 percent for $\frac{K\,\Delta\rho}{Q\,\rho_f} X > 1.7$ (fig. 6). These differences in the interface position are due to (1) errors associated with discretization of the flow system, and (2) differences in model-computed head and the exact head distribution specified by Glover (1959) along the top of the aquifer.



Specifications: $K_x = K_z = 0.025$ ft/s

$\rho_f = 1.000$ g/cm^3, $\rho_s = 1.025$ g/cm^3

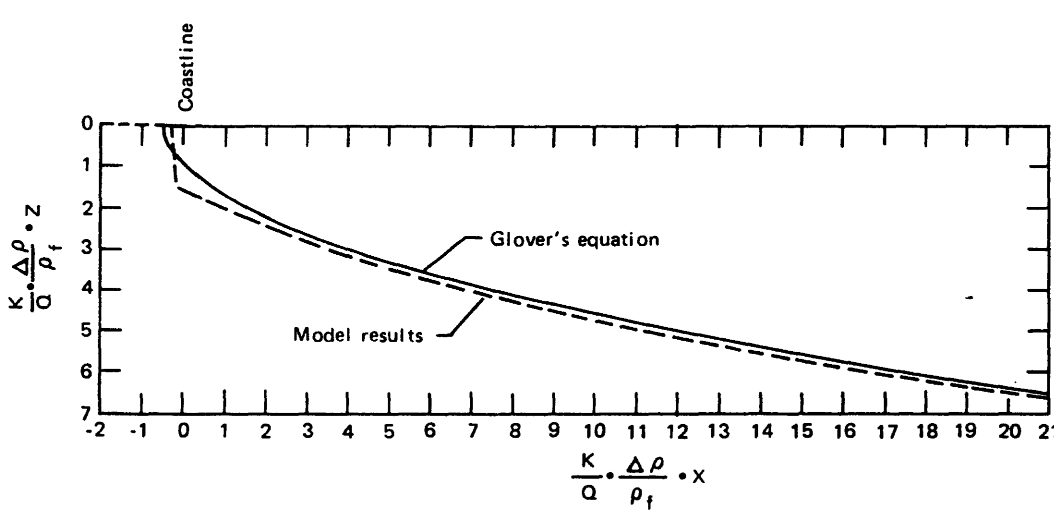
Δ X varies from 250 ft. to 10,000 ft.

Δ Z varies from 250 ft. to 4,000 ft.

No. cells in X-direction (inside model border) = 40

No. cells in Z-direction (model layers) = 25

Figure 5.–Specification of a test problem used to compare model results with Glover's equation.



$\frac{K}{Q} \cdot \frac{\Delta\rho}{\rho_f} \cdot Z$

$\frac{K}{Q} \cdot \frac{\Delta\rho}{\rho_f} \cdot X$

Figure 6.–Comparison of model results with Glover's equation.

USER'S GUIDE FOR THE MODEL

This section describes the preparation of data read by SSIM3D (model input), job control language for running SSIM3D, and the preparation of data and job control language for an example problem.

Preparation of Model Input

Data read by the model are divided into six groups: Group I data define model dimensions, define parameters that control execution of the model, and initialize parameters used for model computations; Group II data are used for continuation of a previous simulation; Group III data are used for locating the position of a sharp interface between freshwater and saltwater; Group IV data redefine pumping rates and other aquifer stresses for each simulated stress period; Group V data are values of parameters and coefficients (see eqs. 1-10) specified for each model cell (array data) at the start of simulation; and Group VI data contain observed heads that are compared with model-computed heads.

Data are read from records stored in computer files, and the record formats are described in this section. Formatted records have a fixed length of 80 columns (or bytes) and unformatted records have variable lengths. In the following descriptions, "Ident" is used to identify a data field and refers to a FORTRAN variable in the model, "Format" refers to the type of data (I for integer, F for floating-point, and A for alphanumeric) and the number of columns in a data field, and "Rt.Col" refers to the rightmost column for a data field. When coding data on records, numeric data must be right-justified in a data field and alphanumeric data must be left-justified. For some of the FORTRAN variables that are dimensioned in the model, the ranges of subscripts shown in parentheses after the variable names refer to word (4-byte or 8-byte word) counts in the dimensioned variables. Subscript notations have been simplified, and sometimes dropped, to make this documentation more readable. The FORTRAN file numbers specified in this documentation refer to files described in "Job Control Language for Running the Model."

Any unit of length [L] may be chosen by the model user, but it must be the same for all parameters. The unit of time [t] must be seconds.

The following explanations for preparing model input are supplemented with explanations given by Trescott (1975), Trescott and Larson (1976), and Torak (1982).

Group I data

These data define model dimensions, define parameters that control execution of the model, and define parameters used for model computations.

All records are formatted and are read from FORTRAN file number 5.

1. Record 1. (Always include this record)
   Ident: HEADNG(1 to 20)
   Format: 20A4
   Rt.Col: 80

   Description:
   HEADNG = Title for a simulation run.

2. Record 2 (continuation of record 1).
   Ident: HEADNG(21 to 33)
   Format: 13A4
   Rt.Col: 52

3. Record 3. (Always include this record)

Ident:	IO	JO	KO	ITMAX	NCH	NRIV	NDRAIN	LWTABL	LRECH	ISALT	MSQCAL
Format:	I5	I5	I5	I5	I5	I5	I5	I5	I5	I5	I5
Rt.Col:	5	10	15	20	25	30	35	40	45	50	55

   Description:
   IO = Number of rows (y-direction) in the model. There must be at least 3 rows.
   JO = Number of columns (x-direction) in the model. There must be at least 3 columns.
   KO = Number of layers (z-direction) in the model. The bottom layer in the model is number 1 and the top layer is KO.
   ITMAX = Number of iterations (maximum of 50) allowed for the SIP equation-solving procedure.
   NCH = Maximum number of constant-head cells in the model. Constant-head cells are located by coding -1 values in the array data set S (see Group V data).
   NRIV = Maximum number of spring cells below sea level plus leaky-river cells above sea level.
   If NRIV >0, the array data sets IDR, RC, RB, and RH are read into the model (see Group V data).
   NDRAIN = Maximum number of spring cells and drain cells above sea level.
   If NDRAIN >0, the array data sets IDN, DRCF, and DREL are read into the model (see Group V data).
   LWTABL = Layer number for the water-table aquifer if <u>WATE</u> is coded for IWATER on record 4 of Group I data. The default value is layer KO. If <u>WATE</u> is not coded for IWATER on record 4, all model layers are assumed to be confined.
   LRECH = Method of applying recharge to the model (must code <u>RECH</u> for IQRE on record 4).
   = 0 to apply recharge to layer LWTABL (or layer KO if LWTABL = 0)
   = -1 to read the array data set LAYRCH (see Group V data) for specifying layer numbers of cells to be recharged.
   = >0 is the number of the lowest layer to be recharged. The

model selects the topmost active cells in layers LRECH through KO and applies recharge to these cells. Select this option if recharge is to be applied to a confined cell located below a water-table cell when a water-table cell goes dry (the confined cell is not converted to a water-table cell).

ISALT = 1 to compute the position of a sharp interface between freshwater and saltwater. Group III data are read into the model.
= 0 to simulate flow of fresh ground water without saltwater.

MSQCAL = Compute the root-mean-square for differences between observed and model-computed heads. Observed head values are read into the model (see Group VI data).
= 0 to skip these computations.
= 1 to perform computations after each stress period.
= 2 to perform computations only after the last stress period.

4. Record 4. (Always include this record)

Ident:	IDRAW	IHEAD	IFLO	IDK1	IDK2	IWATER	IQRE	IPU1	IPU2
Format:	A4,1X	A4,1X	A4,1X	A4,1X	A4,1X	A4,1X	A4,1X	A4,1X	A4,1X
Rt.Col:	4	9	14	19	24	29	34	39	44

Ident:	ITK	IEQN
Format:	A4,1X	A4
Rt.Col:	49	54

Description:

IDRAW = DRAW to print tables for drawdown on FORTRAN file number 8 at the end of each KTH time step (see record 9). Model layers can be selected by coding records 10 and 11. Drawdown is computed by subtracting model-computed head from initial head read at the beginning of simulation (see the data set STRT in Group V data).

IHEAD = HEAD to print tables for computed head on FORTRAN file number 8 at the end of each KTH time step (see record 9). Model layers can be selected by coding records 10 and 11.

IFLO = MASS to print a mass balance on FORTRAN file number 6 at the end of each KTH time step (see record 9). A mass balance is always printed at the end of each stress period. A mass balance will be printed for each model layer if IBALAY = 1 (see record 5).

IDK1 = DK1 to read from FORTRAN file number 4 values of starting head, mass balance parameters, and simulation time for continuation of a previous simulation (see records 4 and 5 in Group II data).
= HEDR to read from FORTRAN file number 4 the values of initial head for the start of a new simulation (simulation time and mass balance parameters are not read and are set to zero). These values replace the values read for the STRT data set (see Group V data).

IDK2 = DK2 to write on FORTRAN file number 4 values of computed head, mass balance parameters, and simulation time at the end of a simulation run (see records 4 and 5 in Group II data). These data can be used in another simulation run either as initial conditions for a new simulation or as starting conditions for continuing a previous similation.

IWATER = WATE if layer LWTABL is a water-table aquifer (unconfined). The array data sets PERM and BOTTOM are read into the model (see Group V data).

IQRE = RECH to read recharge values (see the array data set QRE in Group V data).

IPU1 = PUN1 to read from FORTRAN file number 3 values of starting head, mass balance parameters, and simulation time for continuation of a previous simulation (see records 1, 2, and 3 in Group II data).

IPU2 = PUN2 to write on FORTRAN file number 3 values of computed head, mass balance parameters, and simulation time at the end of simulation (see records 1, 2, and 3 in Group II data).

ITK = ITKR to read values of vertical leakage coefficients between layers (see the array data set TK in Group V data).

IEQN = EQN3 if equation 3 (see Trescott, 1975) is being solved; otherwise the model solves equation 1 (this documentation) or equations 2 and 11 (this documentation).

5. Record 5. (Always include this record)

Ident:	NVSEC	NSUBAL	IBALAY	IPFLX	ISVFLX	ISVHED	ISVHDX
Format:	I5	I5	I5	I5	I5	I5	I5
Rt.Col:	5	10	15	20	25	30	35

Description:

NVSEC = Number of vertical sections for which computed head and saltwater-intrusion codes are printed or saved on disk file. If NVSEC >0, records 6 and 7 will be read.

NSUBAL = Number of model subareas used for mass balance computations. If NSUBAL >0, record 8 will be read. This option is useful for computing mass balance components in subareas of a model when overlapping models are used to simulate flow in extensive aquifers. The mass balance is summed for all subareas.

IBALAY = 1 to print a mass balance for each model layer.

IPFLX = 1 to print flux for each constant-head, leaky-river, and drain cell in a model.

ISVFLX = 1 to save single-precision values of computed flux rates for constant-head, leaky-river, and drain cells at the end of a simulation (flux rates are for the last time step). Unformatted records are written on FORTRAN file number 14.
= 2 to save values of computed flux rates at the end of each time step on FORTRAN file number 14.

ISVHED = 1 to save single-precision values of computed head (plot-file format) at the end of simulation. The value -99 is used for each cell in which transmissivity is zero. One unformatted record for each model layer is written on FORTRAN file number 15.

ISVHDX = 1 to save single-precision values of computed head and intrusion codes (at end of simulation) for selected vertical sections specified when NVSEC >0. One unformatted record for each section is written on FORTRAN file number 16.

6. Record 6. (Include this record only when NVSEC >0)
Records 6 and 7 contain row and column numbers of vertical sections for which model-computed values are printed or saved on disk file. Values on record 6 define the bottom and top model layers for a section, and values on record 7 define the row and column coordinates of each cell on a section line. Each occurrence of record 6 is followed by one or more occurrences of record 7. Records 6 and 7 must be coded for each of the NVSEC vertical sections.

Ident:	LAYSEC(1)	LAYSEC(2)	NSNOD
Format:	I3	I3	I3
Rt.Col:	3	6	9

Description:
LAYSEC(1) = Number of the bottom layer for a vertical section. Remember that layer 1 is at the bottom of the model.
LAYSEC(2) = Number of the top layer for a vertical section.
NSNOD = Number of cells in the x-y plane of a section.

7. Record 7. (Include this record only when NVSEC >0)
Record 7 contains row and column coordinates for each of NSNOD (see record 6) cells located on a section line. Record 7 follows each occurrence of record 6, and record 7 is repeated until NSNOD coordinate pairs are coded for a section. A maximum of 13 coordinate pairs is coded on each occurrence of record 7. The subscript K, shown below, varies from 1 to NSNOD.

Ident:	NODSEC(K,1)	NODSEC(K,2)		NODSEC(K+1,1)	NODSEC(K+1,2)
Format:	I3	I3		I3	I3
Rt.Col:	3	6		9	12

Ident:	NODSEC(K+2,1)	NODSEC(K+2,2)	...	NODSEC(K+12,1)	NODSEC(K+12,2)
Format:	I3	I3		I3	I3
Rt.Col:	15	18		75	78

Description:
NODSEC(K,1) = Row-coordinate for a cell along the line of section.
NODSEC(K,2) = Column-coordinate for a cell along the line of section.

8. Record 8. (Include this record only when NSUBAL >0)
This record contains coordinates of rectangular subareas for which mass balance is computed. The mass balance is summed for all subareas. Record 8 is repeated NSUBAL times.

Ident:	IBT	IBB	IBL	IBR
Format:	I5	I5	I5	I5
Rt.Col:	5	10	15	20

Description:
IBT = Number of the top model row in a subarea.
IBB = Number of the bottom model row in a subarea.
IBL = Number of the left model column in a subarea.
IBR = Number of the right model column in a subarea.

9. Record 9. (Always include this record)

Ident:	NPER	KTH	LENGTH	ERR	ERMULT	WMAX1	BETA1	STEADY	ITRHED
Format:	I5	I5	I5	F5.0	F5.0	F5.0	F5.0	F5.0	I5
Rt.Col:	5	10	15	20	25	30	35	40	45

Ident:	IPHED
Format:	I5
Rt.Col:	50

Description:

NPER = Number of stress periods for this simulation.

KTH = Number of time steps between printouts for a transient simulation. Each printout may consist of head, drawdown, and mass balance (see IHEAD, IDRAW, and IFLO on record 4).

LENGTH = Number of iteration parameters (maximum of 20) for the SIP equation-solving procedure.

ERR = Error criterion for convergence of the SIP equation-solving procedure [L].

ERMULT = Multiplier for ERR when ISALT=1 (see record 3). This multiplication is done after each iteration for locating the interface between freshwater and saltwater.

WMAX1 = Value of the last iteration parameter for the SIP equation-solving procedure (see p. 25 in Trescott, 1975). If zero is coded, the model computes a value for the last iteration parameter. Code 0 ≤WMAX1 <1.

BETA1 = Value of an acceleration (or damping) parameter used for the SIP equation-solving procedure (see p. 26-29 in Trescott, Pinder, and Larson, 1976). If zero is coded, the default value is 1. Code 0 ≤BETA1 <2.

STEADY = Criterion for determining when steady-state conditions exist during a transient simulation of freshwater without saltwater. This criterion is used when solving the transient ground-water flow equation without saltwater intrusion (see eq. 1).

When the ratio of the term $S\frac{\partial H}{\partial t}$ (in eq. 1) to the average of inflow and outflow rates for a time step is less than STEADY, the model assumes a steady-state condition and terminates the current stress period. If zero is coded, the model does not check for steady-state. Code ≤0 STEADY <1.

ITRHED = 1 to print head after each SIP iteration.
= 2 to print head change between two consecutive SIP iterations. Head or head change is printed on FORTRAN file number 10 for model layers selected with the LEVEL2 parameter (see record 11). This option is useful when SIP does not converge because of errors in model input data.
= 0 to print head as defined by IHEAD on record 4.

IPHED = 1 to print initial head read from an input file when continuing a previous simulation (see IDK1 and IPU1 on record 4). Head values are printed on FORTRAN file number 6.

10. Record 10. (Always include this record)
This record contains data needed for printing scaled alphanumeric maps or tabled values of head or drawdown. If alphanumeric maps are to be printed, code XSCALE >0. If tabled values are to be printed, code values for IHEAD or IDRAW on record 4. Both alphanumeric maps and tabled values can be printed for the same simulation run. More information about scaled alphanumeric maps is presented in Trescott (1975, p. II-8). Tabled values of head and drawdown are written on FORTRAN file number 8 and scaled alphanumeric maps are written on file number 9 at the end of each KTH time step (see record 9).

Ident:	XSCALE	YSCALE	DINCH	FACT1	FACT2	MESUR
Format:	F10.0	F10.0	F10.0	F10.0	F10.0	A8
Rt.Col:	10	20	30	40	50	58

Description:

XSCALE = Factor that converts length units in x-direction of the model to length units used on scaled alphanumeric maps (for example, code 5280 to convert from model units of feet to map units of miles).
Code XSCALE >0 to print alphanumeric maps.
Code XSCALE = 0 to omit alphanumeric maps.

YSCALE = Factor that converts length units in y-direction of the model to length units used on scaled alphanumeric maps.
Code YSCALE >0 to print alphanumeric maps or tabled values for all model layers (do not code record 11).
Code YSCALE <0 to read record 11 that specifies model layers for which alphanumeric maps or tabled values are printed.

DINCH = Number of length units per inch (see MESUR) on scaled alphanumeric maps.

FACT1 = Multiplier used to change the magnitude of drawdown values plotted on scaled alphanumeric maps.

FACT2 = Multiplier used to change the magnitude of head values plotted on scaled alphanumeric maps.

MESUR = Name of length unit (see DINCH) for scaled alphanumeric maps.

11. Record 11. (Include this record only when YSCALE <0)
This record specifies layers for which scaled alphanumeric maps or tabled values are printed. Each record column contains data for a model layer, and data are coded in each column to indicate if printout is desired for a model layer.

Ident:	LEVEL1(40)	LEVEL2(40)
Format:	40I1	40I1
Rt.Col:	1,2,...,40	41,42,...,80

Description:

LEVEL1 = 1 to print scaled alphanumeric maps or tabled values for model drawdown.
= 0 to omit alphanumeric maps or tabled values.

LEVEL2 = 1 to print scaled alphanumeric maps or tabled values for model head.
= 0 to omit alphanumeric maps or tabled values.

Group II data

These data are used for initial conditions at the start of a new simulation or as starting conditions for the continuation of a previous simulation. Records 1-3 are formatted on FORTRAN file number 3; records 4 and 5 are unformatted on file number 4. The model creates these records at the end of a simulation run if a value is coded for IPU2 or IDK2 on record 4 of Group I data, or the model reads these records at the start of a simulation run if a value is coded for IPU1 or IDK1 on record 4 of Group I data.

1. Record 1. (Include this record when PUN1 is coded for IPU1 on record 4 of Group I data)

Ident:	SUM	SUMP
Format:	F20.0	F20.0
Rt.Col:	20	40

Description:
- SUM = Time at the end of a simulation (seconds).
- SUMP = Duration of the last stress period for a simulation (seconds).

2. Record 2. (Include this record when PUN1 is coded for IPU1 on record 4 of Group I data.)
Code 6 physical records for each model layer. The first 3 records contain volumes of water added to the model and the second 3 records contain volumes of water withdrawn from the model.

Ident:	CFLX1	CFLX2	CFLX3	CFLX4
Format:	F20.0	F20.0	F20.0	F20.0
Rt.Col:	20	40	60	80

Ident:	CFLX5	CFLX6	CFLX7	CFLX8
Format:	F20.0	F20.0	F20.0	F20.0
Rt.Col:	20	40	60	80

Ident:	CFLX9
Format:	F20.0
Rt.Col:	20

Description:
All of the following parameters are cumulative volumes for a simulation of duration SUM (see record 1).
- CFLX1 = Change in storage within a layer.
- CFLX2 = From recharge.
- CFLX3 = From constant flux sources (or recharge wells).
- CFLX4 = From constant flux sinks (or pumping wells).
- CFLX5 = From flux into or out of constant-head cells.
- CFLX6 = From flux into or out of leaky rivers above sea level and from flux into springs below sea level.
- CFLX7 = From flux into drains and springs above sea level.
- CFLX8 = From vertical flux across top the of a model layer.
- CFLX9 = From vertical flux across the bottom of a model layer.

3. Record 3. (Include this record when PUN1 is coded for IPU1 on record 4 of Group I data)

Code values for all cells in the model starting with row 1 and column 1 in layer 1 of the model. Code the value for column 1 of each model row in the first data field of a new record and code values for each model column in successive data fields on a record. When the last data field on a record is coded and the last value for a model row has not been coded, continue to the first data field on a new record. A total of IO*JO*KO values must be coded.

Ident: PHI(IO,JO,KO)
Format: 8F10.4
Rt.Col: 10,20,...,80

Description:
PHI = Model-computed head values at the end of simulation.

4. Record 4. (Data are read from a file specified on a job control language record for FORTRAN file number 4 when a value is coded for IDK1 on record 4 of Group I data)

Ident: PHI(IO,JO,KO)
Format: Double-precision floating point (unformatted record)

Description:
PHI = Model-computed head values at the end of simulation.

5. Record 5. (Data are read from a file specified on a job control language record for FORTRAN file number 4 when DK1 is coded for IDK1 on record 4 of Group 1 data)

Ident: SUM SUMP CFLX1 CFLX2 CLFX3 CLFX4 CFLX5 CFLX6 CFLX7 CFLX8 CFLX9
Format: SUM and SUMP are single-precision floating point variables, and the other variables are double-precision floating point (unformatted record)

Description:
See records 1 and 2.

Group III data

These data are used for locating the position of a sharp interface between freshwater and saltwater.
Record 1 is formatted and read from FORTRAN file number 5. The array data set DBOT must be read as Group V data.

1. Record 1. (Include this record when ISALT=1 on record 3 of Group I data)
Code 2 physical records.

Ident:	RF	NTAD	ISERCH	LFRESH	ISWDIV	ITSTRT	ISOLAT	ZIMAX	ZDAMP
Format:	F10.0	I5	I5	I5	I5	I5	I5	F5.0	F5.0
Rt.Col:	10	15	20	25	30	35	40	45	50

Ident:	THKMIN	THKPCT	THKFUL	THKRST	IREST	MESG1
Format:	F5.0	F5.0	F5.0	F5.0	I5	I5
Rt.Col:	55	60	65	70	75	80

Ident:	INT1PR	INTVAL	INTMAP	INTSAV	IPEQLB	INTCAL	INTSA2
Format:	I5	I5	I5	I5	I5	I5	I5
Rt.Col:	5	10	15	20	25	30	35

Description:

RF = Density factor used to compute the depth below sea level to a sharp interface between freshwater and saltwater (see eq. 11). This factor is computed from the following formula:
$RF = \rho_f/(\rho_s - \rho_f)$.
If 0 is coded, the default value is 40.

NTAD = Maximum number of saltwater iterations for locating the interface.

ISERCH = Value to indicate how model cells are searched to locate the interface.
= 0 to search all cells in the model between layers 1 and LFRESH.
= 1 to search only the left side of the model (columns 1 through ISWDIV).
= 2 to search only the right side of the model (columns ISWDIV through J0).

LFRESH = Uppermost model layer that can be intruded by saltwater. At least one layer must overly this layer. Do not code LFRESH>K0-1.

ISWDIV = Column number that limits the extent of searching for saltwater intrusion in the left or right side of the model (see ISERCH on this record). The default value is J0.

ITSTRT = Iteration number to start using the ZDAMP parameter.

ISOLAT = Minimum number of active cells that must surround a single active cell. If a cell is not surrounded by more than ISOLAT active cells, the cell is considered totally intruded.
Do not code a value larger than 5.
= -1 to omit this check for isolated cells.

ZIMAX = Maximum difference between the damped and undamped (or equilibrium) interface positions [L]. When this difference is less than ZIMAX and no cells are totally intruded or restored in an iteration, the iteration procedure will stop.

ZDAMP = Multiplier used to dampen interface movement. Code 0≤ABS(ZDAMP) ≤1.
= 0 if damping is not used.
= <0 to use single damping.
= >0 to use double damping.

THKMIN = Minimum freshwater thickness for a partly intruded cell [L]. Both THKMIN and THKPCT are used to determine if a cell should be totally intruded when freshwater thickness is very small. If freshwater thickness is less than AMAX1(THKMIN,(cell thickness)*THKPCT), the cell is totally intruded.

THKPCT = Factor used to compute the minimum freshwater thickness allowed in a partly intruded cell. Code 0≤THKPCT<1. If 0 is coded, the default is 1.E-5. Both THKPCT and THKMIN are used to determine when a cell should be totally intruded (see THKMIN).

THKFUL = Factor used to compute the maximum freshwater thickness allowed in a partly intruded cell. Code 0≤THKFUL<1. If 0 is coded, the default value is computed as 1-10*THKPCT. If freshwater thickness exceeds THKFUL*(cell thickness), the cell contains only freshwater.

THKRST = Factor used to compute the freshwater thickness for restoring a totally intruded cell. Code 0≤THKRST<1. If 0 is coded, the default value is THKPCT*5.

IREST = 1 to allow the restoration of freshwater flow in cells that were totally intruded in a previous iteration. Spring cells below sea level are also restored. Cells are restored when head in contiguous cells builds up and pushes the interface below the tops of the cells.
= 2 to restore all cells except spring cells below sea level.
= 0 to omit the task of restoring totally intruded cells.
Note that totally intruded water-table cells are not restored.

MESG1 = 0 to omit printing coordinates of intruded cells.
= >0 is the maximum number of messages to be printed indicating the coordinates of cells when they are intruded or restored. This option can produce volumes of printout.

INT1PR = 1 to print starting values of intrusion codes for all model cells. These intrusion codes are determined by one of the methods described for the parameter INTCAL (coded on this record).

INTVAL = 1 to print intrusion codes for each model layer containing active cells and print a summary of freshwater thicknesses for partly intruded cells. These intrusion codes are for the last saltwater iteration.

INTMAP = 1 to print a map of greatest depths to the freshwater-saltwater interface and a map of intrusion codes for cells in which the interface is located. Zones of lateral intrusion that are underlain by freshwater are not shown on this map. These maps are for the last saltwater iteration.

INTSAV = 1 to save on FORTRAN file number 17 the greatest depths to the freshwater-saltwater interface computed for the last saltwater iteration (see description for IMTMAP).

IPEQLB = 1 to print a map of differences between the damped and undamped (or equilibrium) interface positions computed for the last saltwater iteration.

INTCAL = 0 to compute initial interface depths and intrusion codes for all model cells at the start of a new simulation by using initial heads. Values for initial heads (see the data set STRT in Group V data) should be large enough to position the interface in the bottom layer of the model. Select this option at the start of a new simulation run.

= 1 to read from FORTRAN file number 18 the starting interface depths for all model cells that were saved on disk file (see INTSA2 on this record) at the end of a previous simulation run.

INTSA2 = 1 to save on FORTRAN file number 18 the computed interface depths and intrusion codes for all model cells at the end of a simulation run. These values can be used in another simulation run either as initial conditions for a new simulation or as starting conditions for continuing a previous simulation.

Group IV data

These data redefine parameters at the start of each stress period. Records 1 and 2 are formatted and read from FORTRAN file number 5.

1. Record 1. (Always read at the start of a stress period)
   One record is required for each stress period and there are NPER stress periods (see record 9 in Group I data).

Ident:	KP	NUMT	TMAX	DELT	CDLT	IWEL	IRECH	IRIV	IHED
Format:	I5	I5	F5.0	F5.0	F5.0	I5	I5	I5	I5
Rt.Col:	5	10	15	20	25	30	35	40	45

   Description:

   KP = Number of this stress period (do not exceed the value of NPER coded on record 9 in Group I data).

   NUMT = Number of time steps in this stress period.

   TMAX = Length of this stress period in days.

   DELT = Length of the initial time step in hours.

   CDLT = Multiplier for DELT (described on p. II-3 of Trescott, 1975).

   IWEL = 1 to read new values of an array data set for constant-flux cells. Record 2 must follow this record when IWEL=1.
   = 0 for the first stress period.

   IRECH = 1 to read new values of an array data set for recharge. Record 2 must follow this record when IRECH=1.
   = 0 for the first stress period.

   IRIV = 1 to read new values of an array data set for head in leaky-river cells.
   Record 2 must follow this record when IRIV=1.
   = 0 for the first stress period.

   IHED = 1 to read new values of an array data set for constant-head cells. Record 2 must follow this record when IHED=1.
   = 0 for the first stress period.

2. Record 2. (Include this record when IWEL>0, IRECH>0, IRIV>0, or IHED>0)
   This record follows record 1 and is repeated for the parameters IWEL, IRECH, IRIV, and IHED. The array data set specified by IWEL is read first and the data set specified by IHED is read last.

   Ident: SETID
   Format: 2A4
   Rt.Col: 8

   Description:

   SETID = Identifier for the array data set to be read as Group V data. This identifier must be the same as the one used for SETNAM on record 2 of Group V data.

Group V Data

These data are used to specify values of parameters and coefficients, shown in equations 1-10, for each cell in a finite-difference model. Data are stored in arrays (or dimensioned variables) used in the computer model, and these arrays are initialized at the start of simulation. Some of the arrays are re-initialized at the start of each stress period (see Group IV data).

To initialize arrays in the model, parameter records (80-column, fixed-length, car-image records) must be prepared for each array data set required for a simulation, and, for multiple-layer data sets, one parameter record is prepared for each layer. Each parameter record contains a data set identifier, a layer number for the model, parameters that specify how the array is initialized in the model, and options for printing the data set or saving it on disk file.

The primary input file for formatted records in this data set is FORTRAN file number 22, messages are printed on FORTRAN file number 6, and array data sets are printed on FORTRAN file number 21. Secondary input and output files (formatted or unformatted) are defined by coding values for IRDSK and IWDSK on record 2. All file numbers must be specified in the job control language (see "Job Control Language for Running the Model").

1. Record 1: Project identification record (Code only 1 record)

Ident:	PROJID
Format:	A8
Rt.Col:	8

Description:
PROJID = A project identifier.

2. Record 2: Parameter Record (Code 1 record for each layer in each data set)

Ident:	SETNAM	LAYR	IRFMT	IRDSK	IDUM	IFACTA	IUPROW	IUPNOD
Format:	A8,2X	I5	I5	I5	I5	I5	I5	I5
Rt.Col:	8	15	20	25	30	35	40	45

Ident:	IPRNT	IWFMT	IWDSK	IDUM	IDUM	FACTA
Format:	I5	I5	I5	I5	I5	F10.0
Rt.Col:	50	55	60	65	70	80

Description:
SETNAM = Identifier for an array data set (see "Description of array data sets" following the description of record 7).
LAYR = Layer number in the model. Code 1 for arrays with the identifiers PERM, BOT, QRE, LAYRCH, DELX, DELY, DELZ.
IRFMT = Format code for reading a data set from the file specified by IRDSK.
= 0 to omit reading a data set.
= 1 to read using format 20F4.0
= 2 to read using format 8F10.0.
= 8 to read an unformatted record.
IRDSK = Number of a FORTRAN file from which array data are read for one model layer. Code a value ≥ 22. If zero is coded, the default value is 22 when IRFMT = 1 or 2; however, no file is read if IRDSK = 0 and IRFMT = 8. The file that is specified

must contain record 4, or records 3 and 4, or record 5 of this data group.

IDUM = Code 0 or leave blank.

IFACTA = 0 to multiply all elements in the array data set by a constant value after reading and updating the data set. The constant is the value specified for FACTA on this record. This operation is not performed if FACTA = 0 or 1.
= 1 to add a constant value to all elements in the array data set after reading and updating the data set. The constant is the value specified for FACTA on this record. This operation is not performed if FACTA = 0.
= 2 to assign a constant value to all elements in the array data set. The constant is the value for FACTA specified on this record. This operation is performed after reading a data set and before updating. If data are not read (IRFMT = 0), this operation is always performed using the value specified for FACTA.

IUPROW = 1 to read updates for groups of cells in selected rows of a model layer. This operation is performed after reading a data set or assigning a constant value to a data set. The updates must follow this record and are coded on record 6. The updates are printed on FORTRAN file number 6.

IUPNOD = to read updates for individual cells in a model layer. This operation is performed after reading a data set or assigning a constant value to a data set. The updates must follow this record and are coded on record 7. The updates are printed on FORTRAN file number 6.

IPRNT = Format for printing values for all cells in a model layer. The print file is FORTRAN file number 21.
= 0 to omit printing.
= 1 to print values using the format 18F7.0.
= 2 to print values using the format 10F12.2.
= 3 to print values using the format 10F12.8.
= 4 to print values using the format 1P10E12.4.
This operation is performed after an array data set is initialized using options specified in columns 16-45 of this record.

IWFMT = Format code for writing a data set on FORTRAN file IWDSK.
= 0 to omit writing.
= 1 to write using format 20F4.0.
= 2 to write using format 8F10.0.
= 3 to write using format 8F10.3.
= 4 to write using format 8F10.7.
= 5 to write using format 1P8E10.3.
= 8 to write an unformatted record.

IWDSK = Number of a FORTRAN file on which array data are written for a model layer. Code a value >22. If zero is coded, data are not written on a disk file. Records 3 and 4 are written if 0<IWFMT<8, or record 5 is written if IWFMT = 8. This operation is performed after an array data set is initialized using options specified in columns 16-45 of this record.
= 0 to omit writing a data set on disk file.

IDUM = Code 0 or leave blank.
IDUM = Code 0 or leave blank.
FACTA = Constant value used to perform operations specified by IFACTA.

3. Record 3. (Include this record on file IRDSK when IRDSK >22 and IRFMT = 1 or 2)

This record must precede record 4 for each model layer when IRDSK >22.

Ident: NL SETDSK
Format: I2 A8
Rt.Col: 2 10

Description:

NL = Number of a model layer (must be the same number coded for LAYR on record 2).

SETDSK = Identifier of an array data set (must be the same identifier coded for SETNAM on record 2).

4. Record 4. (Include this record on file IRDSK when IRDSK $\geq$22 and IRFMT = 1 or 2)

This record must follow record 2 when IRDSK = 22, or it follows record 3 when IRDSK >22.

Ident: A (an array)
Format: 20F4.0 or 8F10.0 (depends on value coded for IRFMT on record 2)

Description:

A = Values for an array data set described in "Description of array data sets".

For arrays indexed by row and column coordinates in the model, code values for all cells in a model layer starting with layer 1. Code the value for column 1 of each model row in the first data field of a new record, and code values for each model column in successive data fields on a record. When the last data field on a record is coded and the last value for a model row has not been coded, continue to the first data field on a new record. The arrays PERM, BOT, QRE, and LAYRCH are coded for only one model layer, a value of 1 is coded for LAYR on record 2, and record 2 is coded once for each array. The arrays DELX, DELY, and DELZ are coded in consecutive data fields on a record (continuation records can be used), a value of 1 is coded for LAYR on record 2, and record 2 is coded once for each array.

5. Record 5. (Include this record on file IRDSK when IRDSK >22 and IRFMT = 8.)

This record is unformatted and is read after record 2. This record is not read from FORTRAN file number 22.

Ident: SETDSK(2) STIME NL NR NC A(an array)
Format: Single-precision floating point (unformatted record)

Description:

SETDSK = Identifier for an array data set (must be the same identifier coded for SETNAM on record 2).

STIME = Simulation time.

NL = Layer number in the model (must be the same number coded for LAYR on record 2).
NR = Number of rows in the model.
NC = Number of columns in the model.
A = Values for an array data set.

6. Record 6. (Include this record when IUPROW = 1.)
This record is used to update one or more cells in a row of a model layer, and this record must follow record 2 or 4 on FORTRAN file number 22. This record can be repeated to update several model rows. Code 0 for KLAY to terminate the updates for a layer.

Ident:	KLAY	IR	JC1	JC2	IOPCOD	IDUM	V(J)
Format:	I5	I5	I5	I5	I5	I5	5F10.0
Rt.Col:	5	10	15	20	25	30	40,50,60,70,80

Ident: V(J) -- continuation record
Format: 8F10.0
Rt.Col: 10,20,...,80

Description:
KLAY = Layer number in the model (same as LAYR on record 2). Code 0 to terminate the updates for a layer.
IR = Model row to be updated.
JC1 = Starting column in row IR.
JC2 = Ending column in row IR.
IOPCOD = 0 to multiply the constant specified for V(1) times values for each element in columns JC1 through JC2.
= 1 to add the constant specified for V(1) to values for each element in columns JC1 through JC2.
= 2 to assign the constant specified for V(1) to each element in columns JC1 through JC2.
= 3 to assign the constants specified for each V(J) to each element in columns JC1 through JC2.
IDUM = Code 0 or leave blank.
V(J) = Values used in operations associated with the value of IOPCOD.

If JC2-JC1+1 exceeds 5 and IOPCOD = 3, code the remaining values of V(J) on continuation records with a format of 8F10.0.

7. Record 7. (Include this record when IUPNOD = 1.)
This record is used to update the value of a single cell in a model layer, and this record must follow record 2, 4, or 6 on FORTRAN file number 22. This record can be repeated to update several cells for a layer. Code 0 for KN to terminate the updates for a layer.

Ident:	KN	IN	JN	IDUM	VAL
Format:	I5	I5	I5	I5	F10.0
Rt.Col:	5	10	15	20	30

Description:
KN = Layer number in the model (same as LAYR on record 2). Code 0 to terminate the updates for a layer.

IN = Row coordinate of the cell to be updated.
JN = Column coordinate of the cell to be updated.
IDUM = Code 0 or leave blank.
VAL = Value assigned to the cell.

8. Description of array data sets.

This section describes the array data that are coded to specify values for cells in the model layers. An alphanumeric identifier is given to each data set and this identifier must be coded for SETNAM on record 2 of this data group. The identifier, in most cases, is the same as the name of a dimensioned variable used in the model. Some data sets require that record 2 be coded for each model layer and other data sets require only one occurrence of record 2. For data sets that require record 2 for each model layer, code the layer number for LAYR on each occurrence of record 2. For data sets that require only one occurrence of record 2, code 1 for LAYR. The data sets described in this section are listed in the order in which they are read into the model. Values for all cells in a layer can be set to a constant or a value can be read for each cell, depending on how record 2 is coded. All of the data sets described by Trescott (1975), and additional data sets required by SSIM3D are described in this section. The parameters IO, JO, KO, NDRAIN, and NRIV used in describing the array data sets are defined in Group I data. The following data sets are always required for a simulation: STRT, S, T, TXTY, DELX, DELY, DELZ, and WELL. The remaining data sets described below are required when the appropriate options are specified in Group I, III, and IV data.

Data set identifier	Number of occurrences for record 2	Description of the data set
STRT	KO	Initial head for the start of a new simulation [L]. Head values read as Group II data will replace these values when HEDR is coded for IDK1 (see record 4 in Group I data). This data set is required for each simulation and there are IO*JO values for each layer.
S	KO	Storage coefficient (or specific yield) for an aquifer [L^3/L^3]. Code values >0 for simulation of transient flow in an aquifer system containing only freshwater using equation 1 (code ISALT = 0 in record 3 of Group I data). Code zero values for simulation of steady flow in either an aquifer system containing only freshwater using equation 2 (code ISALT = 0) or an aquifer system containing freshwater and saltwater using equations 2 and 11 (code ISALT = 1). Code -1 values for constant-head cells used in any simulation. This data set is required for each simulation and there are IO*JO values for each layer.
T	KO	Hydraulic transmissivity of saturated water-bearing material [L^2/t]. The model will assign zero values to cells in row 1, row IO, column 1, and column JO. When a water-table aquifer exists in layer LWTABL (see Group I data), values of T must be coded for layer

LWTABL, but they will be recomputed in the model using values for hydraulic conductivity (see the PERM data set) and saturated thickness. This data set is required for each simulation and there are IO*JO values for each layer.

TXTY	KO	Multipliers for hydraulic transmissivities and vertical leakage coefficients to account for anisotropy in the x-, y-, or z-directions. The three multipliers are uniform within each layer. Do not code zeros for these multipliers. Code the multiplier for the x-direction first, then code multipliers for the y- and the z-directions. If each layer has a constant thickness (Δz), a constant vertical hydraulic conductivity (K_z), and a constant lateral hydraulic conductivity ($K_x = K_y$), and if the vertical leakance coefficients (see the TK data set) are not read but are computed by the model, then code the multiplier for the z-direction as $K_z/(K_x \Delta z)$ for each layer (code K_z/K_x for the water-table aquifer). This data set is required for each simulation and there are 3 values for each layer.
TK	KO-1	Coefficients of vertical leakage between a layer and an overlying adjacent layer [t^{-1}]. If each model layer does not have a constant thickness and a constant hydraulic conductivity, the leakage coefficients can be computed using equation 7 and read into the model. If these coefficients are not read, the model will compute them using values for the anisotropy multipliers in the z-direction (see the TXTY data set). This data set is inclued only when <u>ITKR</u> is coded for ITK on record 4 of Group I data. There are IO*JO values for each model layer.
PERM	1	Hydraulic conductivity of saturated materials in the water-table layer [L/t]. The môdel will assign zero values to cells in row 1, row IO, column 1, and column JO. This data set is included only when <u>WATE</u> is coded for IWATER on record 4 of Group I data. There are IO*JO values in this data set. Code 1 for LAYR on record 2.
BOT	1	Bottom elevation of the water-table layer (code negative values where below sea level), [L]. This data set is included only when <u>WATE</u> is coded for IWATER on record 4 of Group I data. There are IO*JO values in this data set. Code 1 for LAYR on record 2.

QRE	1	Recharge rates (positive values) to the topmost model layers [L/t]. The layers to which recharge is applied depend on the value coded for LRECH on record 3 of Group I data. This data set is included only when <u>RECH</u> is coded for IQRE on record 4 of Group I data. There are IO*JO values in this data set. Code 1 for LAYR on record 2.
LAYRCH	1	Numbers for model layers that are to be recharged. This data set is included only when LRECH = -1 on record 3 of Group I data and <u>RECH</u> is coded for IQRE on record 4 of Group I data. There are IO*JO values in this data set. Code 1 for LAYR on record 2.
DELX	1	Dimensions of model cells in the x-direction (width of each column in a finite-difference grid), [L]. This data set is required for each simulation and there are JO values. Code 1 for LAYR on record 2.
DELY	1	Dimensions of model cells in the y-direction (height of each row in a finite-difference grid), [L]. This data set is required for each simulation and there are IO values. Code 1 for LAYR on record 2.
DELZ	1	Dimensions of model layers in the z-direction (thickness of each layer in a finite-difference grid), [L]. DELZ can be set to 1 for each layer if vertical leakage coefficients are read into the model and if equation 1 or 2 of this documentation or equation 4 of Trescott (1975) is being simulated. This data set is required for each simulation and there are KO values. Code 1 for LAYR on record 2.
WELL	KO	Constant rates of discharge (negative values) or recharge (positive values) for wells in a layer $[L^3/t]$. This data set is required for each simulation and there are IO*JO values for each model layer.
IDN	KO	Locators for springs and drains above sea level. Springs and drains that occur in the same cell must be combined. Code a value of 1 where springs or drains exist and 0 elsewhere. This data set is included only when NDRAIN >0 on record 3 of Group I data. There are IO*JO values for each model layer.
DRCF	KO	Discharge coefficients for springs above sea level (see eq. 8) and for drains above sea level $[L^2/t]$. A combined coefficient must be computed for springs and drains that occur in the same model cell. This data set is included only when NDRAIN >0 on record 3 of Group I data. There are IO*JO values for each model layer.

DREL	KO	Elevation at the bottom of a model layer for each spring cell (see eq. 4) or elevation at the bottom of each drain cell [L]. If the bottom of a spring cell is below sea level and the top is above sea level, code this cell as a spring below sea level (see the IDR data set). This data set is included only when NDRAIN >0 on record 3 of Group I data. There are IO*JO values for each model layer.
IDR	KO	Locators for spring cells below sea level or leaky-river cells above sea level. A leaky river and a spring below sea level must not be located in the same cell. Code a value of 1 where a leaky river exists, 2 where a spring below sea level discharges into a body of saltwater, and 0 elsewhere. This data set is included only when NRIV >0 on record 3 of Group I data. There are IO*JO values for each model layer.
RC	KO	Discharge coefficients for springs below sea level (see eq. 9) and for leaky-river cells above sea level (see eq. 10), $[L^2/t]$. This data set is included only when NRIV >0 on record 3 of Group I data. There are IO*JO values for each model layer.
RB	KO	Top elevations for spring cells below sea level or elevations of river bottoms for leaky-river cells above sea level [L]. Elevations below sea level are negative numbers. This data set is included only when NRIV >0 on record 3 of Group I data. There are IO*JO values for each model layer.
RH	KO	Freshwater heads in the saltwater zone for spring cells below sea level (see eq. 5) or water-surface elevations for leaky-river cells above sea level (see eq. 6) [L]. To compute freshwater head for a spring cell that discharges freshwater into the ocean, equation 11 can be rewritten as $RH \doteq \frac{\rho_s - \rho_f}{\rho_f} ZT$ where ZT = -RB. This data set is included only when NRIV >0 on record 3 of Group I data. There are IO*JO values for each model layer.
DBOT	KO	Depth below sea level (positive downward) to the bottom of each model cell [L]. Depths must be coded for model layers 1 through LFRESH+1 (see record 1 in Group III data), and depths can be zero for layers LFRESH+2 to KO. If a cell bottom is above sea level, code a negative value. This data set is included only when ISALT >0 on record 3 of Group I data. There are IO*JO values for each model layer.

Group VI Data

These data are used to compute differences and the root-mean-square for differences between observed and model-computed heads. Records 1 and 2 are formatted and read from FORTRAN file number 41; records 3 and 4 are formatted and read from FORTRAN file number 42. Include this data group only when MSQCAL >0 on record 3 of Group I data.

1. Record 1.
   Dimensions of a regional model grid (this can be larger than the grid dimensions for a subregional model or the two grids can be the same).

Ident:	NROW	NCOL	NLAY
Format:	I5	I5	I5
Rt.Col:	5	10	15

   Description:
   - NROW = Number of rows in the regional model grid.
   - NCOL = Number of columns in the regional model grid.
   - NLAY = Number of layers in both the regional and subregional model grids.

2. Record 2.
   Coordinates of the rectangular subregion for this subregional model (coordinates are referenced to the regional model grid). If coordinates are identical for the regional and subregional model grids, code zeros on this record.

Ident:	NRA1	NRA2	NCA1	NCA2
Format:	I5	I5	I5	I5
Rt.Col:	5	10	15	20

   Description:
   - NRA1 = Row number in the regional grid for the first row in this subregional model.
   - NRA2 = Row number in the regional grid for the last row in this subregional model.
   - NCA1 = Column number in the regional grid for the first column in this subregional model.
   - NCA2 = Column number in the regional grid for the last column in this subregional model.

3. Record 3.
   Model layer in which there are observation points.
   Coordinates of observation points are coded on record 4.

Ident:	K	SETID
Format:	I2	A8
Rt.Col:	2	10

   Description:
   - K = Layer number for all observations coded on record 4.
   - SETID = Identifier for the observations (code something meaningful).

Record 3 is coded once for each model layer and is followed by multiple occurrences of record 4. For multiple layers, record 3 follows record 4 (for the previous layer) that contains 5 zeros.

4. Record 4.
   Data for observation points in layer K (layer is specified on record 3).

   Ident: I J DI DJ HOBS
   Format: Free-field (a blank field must follow each number)

   Description:

   - I = Row coordinate (regional coordinate) for a cell in which the observation point exists.
   - J = Column coordinate (regional coordinate) for a cell in which the observation point exists.
   - DI = Displacement in the row direction [L] from the center of a cell to the observation point. Code a negative value if displacement is in the direction of row 1.
   - DJ = Displacement in the column direction [L] from the center of a cell to the observation point. Code a negative value if displacement is in the direction of column 1.
   - HOBS = Observed value of head at the observation point [L].

   Following the last occurrence of this record for layer K, code a record with zero for each of the five parameters. This zero record can be followed by record 3 for another layer.

Job Control Language for Running the Model

The computer model is programmed to run on the U.S. Geological Survey's AMDAHL 470/V7* computer in Reston, Va., with a FORTRAN 77 compiler and an IBM OS/VS2 operating system. An example of job control language used for running a load module for the model on the AMDAHL V7 is shown below:

```
//jobid JOB (accounting data)
/*LOGONID record if needed
/*PASSWORD record if needed
/*JOBPARM L=20,K=64
/*ROUTE record if needed.
//*$$FILE record if needed.
//GWMODL EXEC FORTVG,ULIB ='userid.LOADLIB',
//      PARM.GO='EP=MAIN,SIZE=4100K,MAP',REGION.GO=4100K,TIME.GO=3
//GO.SYSLIN DD DSN=userid.LOADLIB(SSIM3D),DISP=SHR
//GO.SYSIN DD DSN=userid.GRP134,DISP=SHR
//GO.FT04F001 DD DSN=userid.HEADCALC,DISP=SHR,DCB=BUFNO=2
//GO.FT08F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=9310)
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=9310)
//GO.FT10F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=9310)
//GO.FT13F001 DD UNIT=SYSDA,SPACE=(TRK,(5,5)),
//        DCB=(RECFM=FB,LRECL=133,BLKSIZE=2926,BUFNO=2)
//GO.FT14F001 DD DSN=userid.FLUX,DISP=SHR,DCB=BUFNO=2
//GO.FT15F001 DD DSN=userid.PLTHEAD,DISP=SHR,DCB=BUFNO=2
//GO.FT16F001 DD DSN=userid.XSEC,DISP=SHR,DCB=BUFNO=2
//GO.FT17F001 DD DSN=userid.PLTINTF,DISP=SHR,DCB=BUFNO=2
//GO.FT18F001 DD DSN=userid.INTFSAV,DISP=SHR,DCB=BUFNO=2
//GO.FT21F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=9310)
//GO.FT22F001 DD DSN=userid.GRP5,DISP=SHR
//GO.FT23F001 DD DSN=userid.DXDYDZ,DISP=SHR
//GO.FT24F001 DD DSN=userid.TXTY,DISP=SHR
//GO.FT25F001 DD DSN=userid.STRTHEAD,DISP=SHR
//            DD DSN=userid.STORCOF,DISP=SHR
//            DD DSN=userid.TRANSMIS,DISP=SHR
//            DD DSN=userid.VLKCOF,DISP=SHR
//            DD DSN=userid.RECHARGE,DISP=SHR
//            DD DSN=userid.CONSTFLX,DISP=SHR
//            DD DSN=userid.DRAINS,DISP=SHR
//            DD DSN=userid.SPRINGS,DISP=SHR
//            DD DSN=userid.DBOT,DISP=SHR
//GO.FT41F001 DD DSN=userid.MSQDATA,DISP=SHR
//GO.FT42F001 DD DSN=userid.HEADOBS,DISP=SHR
//
```

---

*Any use of trade names is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey.

The files used in the above job control language are described below:

Name of DD record	FORTRAN file No.	Type of file	Description
SYSLIN	--	U	Contains the compiled FORTRAN code for the model and the IBM FORTRAN library routines used by the model. The data set "userid.LOADLIB" must be partitioned and program SSIM3D is a member in this data set.
FT03F001	3	F	Contains model-computed head and mass-balance values (records 1 to 3 of Group II data) for the end of a previous simulation (see IPU1 on record 4 of Group I data) or computed head and mass-balance values at the end of this simulation (see IPU2 on record 4).
FT04F001	4	U	Contains model-computed head and mass balance values (records 4 and 5 of Group II data) for the end of a previous simulation (see IDK1 on record 4 of Group I data) or computed head and mass balance values at the end of this simulation (see IDK2 on record 4). Values are in double-precision format.
SYSIN	5	F	Contains Group I, III, and IV data.
FT06F001	6	F	Contains messages and printout of some model results (this is included in the cataloged procedure FORTVG).
FT08F001	8	F	Contains printout of computed drawdown (see IDRAW on record 4 of Group I data) or computed head (see IHEAD on record 4).
FT09F001	9	F	Contains printout of scaled alphanumeric maps (see record 10 of Group I data).
FT10F001	10	F	Contains printout of computed head for each iteration in the equation-solving routine (see ITRHED on record 9 of Group I data).
FT13F001	13	F	Temporary file used by subroutine CHECKI when IPFLX>0 (see record 5 in Group I data).
FT14F001	14	U	Contains computed flux for constant-head, drain, and leaky-river cells (see ISVFLX on record 5 of Group I data).
FT15F001	15	U	Contains computed head in plot-file format (see ISVHED on record 5 of Group I data).
FT16F001	16	U	Contains computed head and intrusion codes for vertical sections (see ISVHDX on record 5 of Group I data).
FT17F001	17	U	Contains computed depths to the interface between freshwater and saltwater (see INTSAV on record 1 of Group III data).
FT18F001	18	U	Contains computed interface depths and intrusion codes for model cells (see INTSA2 on record 1 of Group III data).
FT21F001	21	F	Contains printout of array values for Group V data.

FT22F001	22	F	Contains records 1, 2, and selected records for Group V data.
FT23F001	23	F	Contains the Group V data sets DELX, DELY, and DELZ (this file number must be specified for IRDSK on record 2 of Group V data).
FT24F001	24	F	Contains the Group V data set TXTY (this file number must be specified for IRDSK on record 2 of Group V data).
FT25F001	25	U	Contains the Group V data sets STRT, S, T, TK, QRE, WELL, IDN, DRCF, DREL, IDR, RC, RB, RH, DBOT (this file number must be specified for IRDSK on record 2 of Group V data).
FT41F001	41	F	Contains records 1 and 2 of Group VI data.
FT42F002	42	F	Contains records 3 and 4 of Group VI data.

where the file types are defined as

U = unformatted sequential file,
F = formatted sequential file.

The region size of 4100K shown on the above EXEC record is required for a model run that simulates saltwater intrusion and has dimensions of 77 rows, 38 columns, 15 layers, contains 1,690 spring cells below sea level, and 760 spring cells above sea level (dimension of Y vector in MAIN routine is 980100). The region size can be computed as

$$RS = PS + YV + FL + BS$$

where

RS = Region size in kilobytes (1K = 1024 bytes).
PS = Size of compiled FORTRAN code (includes COMMON); approximately 120K (includes 14K for COMMON). The code was compiled with the IBM VS FORTRAN compiler using level 1 optimization.
YV = Size of the Y vector in the MAIN routine; computed as 3.91K for each 1000 elements (4-byte words) in the vector.
FL = Size of the IBM FORTRAN library; approximately 40K.
BS = Buffer sizes for files; for each file, the size is computed as the block size times the number of buffers assigned to the file.

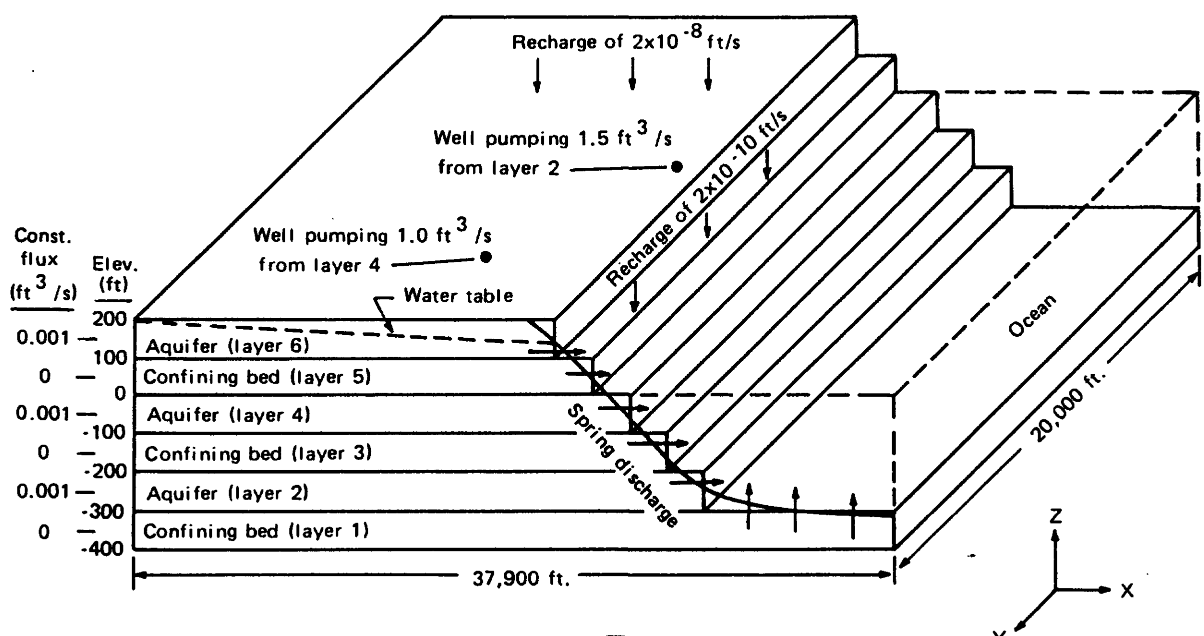
The size of the Y vector in the MAIN routine and the size needed for a simulation are printed out when the model is run. If the Y vector is too small, the job is aborted.

Example Problem

For an example problem (illustrated in figure 7), the model is used to compute the position of a sharp interface between moving fresh ground water and stationary seawater. A constant flux boundary is specified along the left side of the flow system; zero flux boundaries are specified along the front, back, and bottom; recharge is specified for the layers above sea level; springs above sea level are specified where layers intersect land surface; and springs below sea level are specified where layers intersect the ocean bottom. The low-permeability confining beds are specified as layers because they can be intruded by seawater. There are two pumping wells in this problem. The dimensions of the finite-difference grid and the hydraulic properties of the flow system are shown in figure 7.

The job control language and a listing of model-input data are shown in table 1. The formats for the data records are described elsewhere in this report. Where numeric fields are left blank, zero values are assumed by the model. None of the records in table 1 are totally blank.

Table 2 contains printout from a simulation run using data shown in table 1.



Specifications: $K_x = K_y = 2\times10^{-4}$ ft/s, $K_z = 2\times10^{-5}$ ft/s — Aquifers

$K_x = K_y = 2\times10^{-6}$ ft/s, $K_z = 2\times10^{-8}$ ft/s — Confining beds

$\rho_f = 1.000$ g/cm^3, $\rho_s = 1.025$ g/cm^3

ΔX varies from 1500 to 2000 ft., $\Delta Y = 2000$ ft., $\Delta Z = 100$ ft.

No. cells in X-direction (inside model border) = 20

No. cells in Y-direction (inside model border) = 10

Figure 7.–Specification of an example problem for the model.

REFERENCES CITED

Bear, J., 1979, Hydraulics of Groundwater: New York, McGraw-Hill Inc., 569 p.

Glover, R. E., 1959, The pattern of fresh-water flow in a coastal aquifer: Journal of Geophysical Research, v. 64, no. 4, p. 457-459.

Guswa, J. H., and Le Blanc, D. R., 1981, Digital models of ground-water flow in the Cape Cod aquifer system, Massachusetts: U.S. Geological Survey Open-File Report 80-67, 128 p.

Hantush, M.S., 1960, Modification of the theory of leaky aquifers: Journal of Geophysical Research, v. 65, no. 11, p. 3713-3725.

Hubbert, M. K., 1940, The theory of ground-water motion: The Journal of Geology, v. 48, no. 8, p. 858-870.

Mualem, Y., and Bear, J., 1974, The shape of the interface in steady flow in a stratified aquifer: Water Resources Research, v. 10, no. 6, p. 1207-1215.

Torak, L.J, 1982, Modifications and corrections to the finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Water Resources Investigations Report 82-4052, 63 p.

Trescott, P.C., 1975, Documentation of a finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Open-File Report 75-438, 103 p.

Trescott, P.C., and Larson, S.P., 1976, Supplement to documentation of finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Open-File Report 76-591, 21 p.

Trescott, P.C., Pinder, G.F., and Larson, S.P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey, Techniques of Water-Resources Investigations, Book 7, Chapter C1, 116 p.

TABLE 1.--Model input data for the example problem shown in figure 7

```
//AG40XMLS JOB (nnnnnnnnn,DS00),'SAPIK',CLASS=C
//*** EXAMPLE PROBLEM FOR MODEL DOCUMENTATION ***
/*JOBPARM L=20,K=64
//GWMODL EXEC FORTVG,ULIB='AG40XML.LOADLIB',
// PARM.GO='MAP,EP=MAIN,SIZE=500K',REGION.GO=500K,TIME.GO=1
//GO.SYSLIN    DD DSN=AG40XML.LOADLIB(SSIM3D),DISP=SHR,DCB=BUFNO=2
//GO.FT08F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=9310)
//GO.SYSIN    DD *  GROUP I, III, IV DATA
EXAMPLE PROBLEM FOR ILLUSTRATING THE USE OF THE THREE-DIMENSIONAL SEAWATER INTRU
SION MODEL
   12   22    6   20    0   70   20    6    5    1    0
     HEAD                    WATE RECH
    2    0    1    1
  1  6 22
  4  1  4  2  4  3  4  4  4  5  4  6  4  7  4  8  4  9  4 10  4 11  4 12  4 13
  4 14  4 15  4 16  4 17  4 18  4 19  4 20  4 21  4 22
  1  6 22
  7  1  7  2  7  3  7  4  7  5  7  6  7  7  7  8  7  9  7 10  7 11  7 12  7 13
  7 14  7 15  7 16  7 17  7 18  7 19  7 20  7 21  7 22
    1    1    5   .2   .9 .998   .9    0    0    0
         0         -1
                                               010101
        40   30    0    4    0    1    1    2   .3   .5  .01  .99  .02    1    0
    1    1    1    0    1    0    0
    1    1    1   24    1    0    0    0    0
//GO.FT21F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=9310)
//GO.FT22F001 DD *  GROUP V DATA
EXAMPLE
STRT          1                   2            1                            110
STRT          2                   2                                         120
STRT          3                   2                                         130
STRT          4                   2                                         140
STRT          5                   2                                         180
STRT          6                   2                                         200
S             1                                2
S             2
S             3
S             4
S             5
S             6
T             1                   2                                       2.E-4
T             2    1   22         0            3                          2.E-2
   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
```

```
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0
   0
   0
   0
T           3  1  22       0                             2.E-4
   0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0
   0
   0
   0
T           4  1  22       0                             2.E-2
   0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
   0
   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0
```

```
   0
   0
   0
T           5    1   22          0                                       2.E-4
   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0
   0
   0
   0
T           6                    2                                           0
TXTY        1    2   22                        4
      1          1     1.E-4
TXTY        2    2   22
      1          1     1.E-3
TXTY        3    2   22
      1          1     1.E-4
TXTY        4    2   22
      1          1     1.E-3
TXTY        5    2   22
      1          1     1.E-4
TXTY        6    2   22
      1          1     1.E-1
PERM        1    1   22          0             3                             2.E-4
   0
   0
   0   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
   0
   0   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
   0
   0   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
   0
   0   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
   0
   0   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
   0
   0
   0
BOT           1                  2              2                        100.0
QRE           1    1   22        0              4                        2.0E-8
   0
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0   1   1   1   1   1   1   1   1   1   1   1   1   1 .01
   0
   0
   0
DELX          1    2   22                       2
      2000      2000      2000      2000      2000      2000      2000      2000
      2000      2000      2000      2000      2000      1700      1500      1500
      1500      1700      2000      2000      2000      2000
DELY          1                  2                                        2000
DELZ          1                  2                                         100
WELL          1
WELL          2                  2         1    2                            0
    2    4   12         -1.5
    2    2    2        0.001
    2    3    2        0.001
    2    4    2        0.001
    2    5    2        0.001
    2    6    2        0.001
    2    7    2        0.001
    2    8    2        0.001
    2    9    2        0.001
    2   10    2        0.001
    2   11    2        0.001
    0    0    0
WELL          3
WELL          4                  2         1    2                            0
    4    7    6         -1.0
    4    2    2        0.001
```

```
    4    3    2           0.001
    4    4    2           0.001
    4    5    2           0.001
    4    6    2           0.001
    4    7    2           0.001
    4    8    2           0.001
    4    9    2           0.001
    4   10    2           0.001
    4   11    2           0.001
    0    0    0
WELL        5
WELL        6                     2          1    2                              0
    6    2    2           0.001
    6    3    2           0.001
    6    4    2           0.001
    6    5    2           0.001
    6    6    2           0.001
    6    7    2           0.001
    6    8    2           0.001
    6    9    2           0.001
    6   10    2           0.001
    6   11    2           0.001
    0    0    0
IDN         1
IDN         2
IDN         3
IDN         4
IDN         5                     2          1    2                              0
    5    2   15               1
    5    3   15               1
    5    4   15               1
    5    5   15               1
    5    6   15               1
    5    7   15               1
    5    8   15               1
    5    9   15               1
    5   10   15               1
    5   11   15               1
    0    0    0
IDN         6                     2          1    2                              0
    6    2   14               1
    6    3   14               1
    6    4   14               1
    6    5   14               1
    6    6   14               1
    6    7   14               1
    6    8   14               1
    6    9   14               1
    6   10   14               1
    6   11   14               1
    0    0    0
DRCF        1
DRCF        2
DRCF        3
DRCF        4
DRCF        5                     2               4                         5.3E-4
DRCF        6                     2                                         4.7E-2
DREL        1                     2                                              0
```

```
DREL      2            2                                0
DREL      3            2                                0
DREL      4            2                                0
DREL      5            2                                0
DREL      6            2          2                   100
IDR       1   1   22              2
 0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2
 2  0
 0
 0
IDR       2   1   22
 0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
 0
 0
 0
IDR       3   1   22
 0
 0
```

```
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   0
   0
   0
IDR            4    1   22
   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   0
   0
   0
IDR            5
IDR            6
RC             1                   2              4                      1.6E-3
RC             2                   2                                     4.7E-2
RC             3                   2                                     5.3E-4
RC             4                   2                                     5.3E-2
RC             5                   2                                          0
RC             6                   2                                          0
RB             1                   2              2                        -300
RB             2                   2                                       -200
RB             3                   2                                       -100
```

RB	4	2		0
RB	5			
RB	6			
RH	1	2	2	7.5
RH	2	2		5.0
RH	3	2		2.5
RH	4	2		0
RH	5			
RH	6			
DBOT	1	2	2	400.0
DBOT	2	2		300.0
DBOT	3	2		200.0
DBOT	4	2		100.0
DBOT	5	2		0
DBOT	6	2		-100
//				

TABLE 2.--Printout from a simulation run for the example problem shown in figure 7

```
EXAMPLE PROBLEM FOR ILLUSTRATING THE USE OF THE THREE-DIMENSIONAL SEAWATER INTRUSION MODEL

DATA FROM RECORD 3 IN GROUP I DATA:
-----------------------------------
     ROWS IN MODEL                                    =   12
     COLUMNS IN MODEL                                 =   22
     LAYERS IN MODEL                                  =    6
     MAX. ITERATIONS FOR EQUATION-SOLVING PROCEDURE   =   20
     MAX. NO. OF CONSTANT-HEAD CELLS                  =    0
     LEAKY RIVER CELLS                                =   70
     DRAIN CELLS                                      =   20
     LAYER FOR WATER-TABLE AQUIFER                    =    6
     METHOD OF INCLUDING RECHARGE                     =    5
     SOLVE FOR SALTWATER INTERFACE                    =    1
     COMPUTE ROOT-MEAN-SQUARE FOR HEAD DIFFERENCES    =    0

DATA FROM RECORD 4 IN GROUP I DATA:
-----------------------------------
     IDRAW  =
     IHEAD  = HEAD
     IFLO   =
     IDK1   =
     IDK2   =
     IWATER = WATE
     IQRE   = RECH
     IPU1   =
     IPU2   =
     ITK    =
     IEQN   =

*** WORDS USED FOR Y-VECTOR IN MAIN ROUTINE =    36572

    DIMENSIONED SIZE FOR Y-VECTOR           =    50000

DATA FROM RECORD 5 IN GROUP I DATA:
-----------------------------------
     NVSEC  =     2
     NSUBAL =     0
     IBALAY =     1
     IPFLX  =     1
     ISVFLX =     0
     ISVHED =     0
     ISVHDX =     0

DATA FROM RECORDS 6 AND 7 IN GROUP I DATA: VERTICAL SECTIONS FOR WHICH MODEL PARAMETERS ARE PRINTED
----------------------------------------------------------------------------------------------------
    TOTAL NO. SECTIONS =   2

    SECTION NO. =   1    BOTTOM LAYER =   1    TOP LAYER =   6    NO. CELLS ALONG SECTION =   22

    ............................ROW AND COLUMN COORDINATES (I,J) FOR EACH LAYER OF THIS SECTION............................
    ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J )
       4, 1      4, 2      4, 3      4, 4      4, 5      4, 6      4, 7      4, 8      4, 9      4, 10     4, 11
       4, 12     4, 13     4, 14     4, 15     4, 16     4, 17     4, 18     4, 19     4, 20     4, 21     4, 22

    SECTION NO. =   2    BOTTOM LAYER =   1    TOP LAYER =   6    NO. CELLS ALONG SECTION =   22

    ............................ROW AND COLUMN COORDINATES (I,J) FOR EACH LAYER OF THIS SECTION............................
    ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J ) ( I , J )
       7, 1      7, 2      7, 3      7, 4      7, 5      7, 6      7, 7      7, 8      7, 9      7, 10     7, 11
       7, 12     7, 13     7, 14     7, 15     7, 16     7, 17     7, 18     7, 19     7, 20     7, 21     7, 22
```

```
DATA FROM RECORD 9 IN GROUP I DATA:
-----------------------------------
     NPER   =     1
     KTH    =     1
     LENGTH =     5
     ERR    =     0.19999999
     ERMULT =     0.89999998
     WMAX1  =     0.99800003
     BETA1  =     0.89999998
     STEADY =     0.0
     ITRHED =     0
     IPHED  =     0

DATA FROM RECORDS 10 AND 11 IN GROUP I DATA: PARAMETERS FOR PRINTING MAPS
-------------------------------------------------------------------------
     CONVERSION FACTOR FOR LENGTH UNITS IN X-DIRECTION OF SCALED MAPS (XSCALE) =      0.0
     CONVERSION FACTOR FOR LENGTH UNITS IN Y-DIRECTION OF SCALED MAPS (YSCALE) =     -1.0000
     NUMBER OF LENGTH UNITS PER INCH ON SCALED MAPS (DINCH)                    =      0.0
     MULTIPLIER FOR DRAWDOWN PLOTTED ON SCALED MAPS (FACT1)                    =      1.0000
     MULTIPLIER FOR HEAD PLOTTED ON SCALED MAPS (FACT2)                        =      1.0000
     NAME OF LENGTH UNIT FOR SCALED MAPS (MESUR)                               =

     PRINT HEAD VALUES (OR MAPS) FOR THE FOLLOWING LAYERS:

        0  2  0  4  0  6

---------------------------------------------------------
INITIALIZE SUBROUTINE GWDBAS FOR PROCESSING GROUP V DATA
---------------------------------------------------------

     PROJECT IDENTIFIER             = EXAMPLE
     ACCESS CODE FOR DATA BASE DISK =     0
     MODEL DIMENSIONS:  LAYERS      =     6
                        ROWS        =    12
                        COLUMNS     =    22

INITIALIZE CELLS FOR DATA SET = STRT    , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.10000E+02

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = STRT    , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.20000E+02

INITIALIZE CELLS FOR DATA SET = STRT    , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.30000E+02

INITIALIZE CELLS FOR DATA SET = STRT    , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.40000E+02

INITIALIZE CELLS FOR DATA SET = STRT    , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.80000E+02

INITIALIZE CELLS FOR DATA SET = STRT    , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  2.00000E+02

INITIALIZE CELLS FOR DATA SET = S       , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0
```

```
   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = S       , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = S       , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = S       , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = S       , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = S       , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = T       , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  2.00000E-04

INITIALIZE CELLS FOR DATA SET = T       , LAYER =  2:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONSTANT =  2.00000E-02

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = T       , LAYER =  3:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONSTANT =  2.00000E-04

INITIALIZE CELLS FOR DATA SET = T       , LAYER =  4:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONSTANT =  2.00000E-02

INITIALIZE CELLS FOR DATA SET = T       , LAYER =  5:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONSTANT =  2.00000E-04

INITIALIZE CELLS FOR DATA SET = T       , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = TXTY    , LAYER =  1:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = TXTY    , LAYER =  2:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.
```

```
INITIALIZE CELLS FOR DATA SET = TXTY    , LAYER =  3:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = TXTY    , LAYER =  4:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = TXTY    , LAYER =  5:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = TXTY    , LAYER =  6:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = PERM    , LAYER =  1:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONSTANT =  2.00000E-04

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = BOT     , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.00000E+02

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = QRE     , LAYER =  1:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONSTANT =  2.00000E-08

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = DELX    , LAYER =  1:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = DELY    , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  2.00000E+03

INITIALIZE CELLS FOR DATA SET = DELZ    , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.00000E+02

INITIALIZE CELLS FOR DATA SET = WELL    , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = WELL    , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0
```

READ FROM CARD-IMAGE FILE (FILE NO. 22) UPDATES FOR SELECTED CELLS IN LAYER.

ROW	COL	VALUE
4	12	-1.50000E+00
2	2	1.00000E-03
3	2	1.00000E-03
4	2	1.00000E-03
5	2	1.00000E-03
6	2	1.00000E-03
7	2	1.00000E-03
8	2	1.00000E-03
9	2	1.00000E-03
10	2	1.00000E-03
11	2	1.00000E-03

PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = WELL , LAYER = 3:

SET ALL CELLS IN LAYER TO CONSTANT = 0.0

INITIALIZE CELLS FOR DATA SET = WELL , LAYER = 4:

SET ALL CELLS IN LAYER TO CONSTANT = 0.0

READ FROM CARD-IMAGE FILE (FILE NO. 22) UPDATES FOR SELECTED CELLS IN LAYER.

ROW	COL	VALUE
7	6	-1.00000E+00
2	2	1.00000E-03
3	2	1.00000E-03
4	2	1.00000E-03
5	2	1.00000E-03
6	2	1.00000E-03
7	2	1.00000E-03
8	2	1.00000E-03
9	2	1.00000E-03
10	2	1.00000E-03
11	2	1.00000E-03

PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR OATA SET = WELL , LAYER = 5:

SET ALL CELLS IN LAYER TO CONSTANT = 0.0

INITIALIZE CELLS FOR DATA SET = WELL , LAYER = 6:

SET ALL CELLS IN LAYER TO CONSTANT = 0.0

READ FROM CARD-IMAGE FILE (FILE NO. 22) UPDATES FOR SELECTED CELLS IN LAYER.

ROW	COL	VALUE
2	2	1.00000E-03
3	2	1.00000E-03
4	2	1.00000E-03
5	2	1.00000E-03
6	2	1.00000E-03
7	2	1.00000E-03
8	2	1.00000E-03
9	2	1.00000E-03
10	2	1.00000E-03
11	2	1.00000E-03

PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

```
*** COUNT OF CONSTANT-HEAD CELLS =     0  (USER SPECIFICATION =     0) ***

INITIALIZE CELLS FOR DATA SET = IDN     , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = IDN     , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = IDN     , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = IDN     , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = IDN     , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

   READ FROM CARD-IMAGE FILE (FILE NO. 22) UPDATES FOR SELECTED CELLS IN LAYER.

      ROW  COL    VALUE
        2   15   1.00000E+00
        3   15   1.00000E+00
        4   15   1.00000E+00
        5   15   1.00000E+00
        6   15   1.00000E+00
        7   15   1.00000E+00
        8   15   1.00000E+00
        9   15   1.00000E+00
       10   15   1.00000E+00
       11   15   1.00000E+00

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = IDN     , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

   READ FROM CARD-IMAGE FILE (FILE NO. 22) UPDATES FOR SELECTED CELLS IN LAYER.

      ROW  COL    VALUE
        2   14   1.00000E+00
        3   14   1.00000E+00
        4   14   1.00000E+00
        5   14   1.00000E+00
        6   14   1.00000E+00
        7   14   1.00000E+00
        8   14   1.00000E+00
        9   14   1.00000E+00
       10   14   1.00000E+00
       11   14   1.00000E+00

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

*** COUNT OF DRAIN CELLS =    20  (USER SPECIFICATION =    20) ***

INITIALIZE CELLS FOR DATA SET = DRCF    , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0
```

```
INITIALIZE CELLS FOR DATA SET = DRCF    , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DRCF    , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DRCF    , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DRCF    , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  5.30000E-04

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = DRCF    , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  4.70000E-02

INITIALIZE CELLS FOR DATA SET = DREL    , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DREL    , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DREL    , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DREL    , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DREL    , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DREL    , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.00000E+02

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = IDR     , LAYER =  1:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = IDR     , LAYER =  2:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = IDR     , LAYER =  3:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = IDR     , LAYER =  4:

   READ FROM CARD-IMAGE FILE (FILE NO. 22) VALUES FOR ALL CELLS IN LAYER.
```

```
INITIALIZE CELLS FOR DATA SET = IDR      , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = IDR      , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

*** COUNT OF LEAKY-RIVER CELLS =    60  (USER SPECIFICATION =    70) ***

INITIALIZE CELLS FOR DATA SET = RC       , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.60000E-03

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = RC       , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  4.70000E-02

INITIALIZE CELLS FOR DATA SET = RC       , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  5.30000E-04

INITIALIZE CELLS FOR DATA SET = RC       , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  5.30000E-02

INITIALIZE CELLS FOR DATA SET = RC       , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RC       , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RB       , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT = -3.00000E+02

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = RB       , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT = -2.00000E+02

INITIALIZE CELLS FOR DATA SET = RB       , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT = -1.00000E+02

INITIALIZE CELLS FOR DATA SET = RB       , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RB       , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RB       , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RH       , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  7.50000E+00
```

```
   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = RH       , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  5.00000E+00

INITIALIZE CELLS FOR DATA SET = RH       , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  2.50000E+00

INITIALIZE CELLS FOR DATA SET = RH       , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RH       , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = RH       , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

DATA FROM RECORD 1 IN GROUP III DATA: PARAMETERS FOR THE SEAWATER INTRUSION PROBLEM
-------------------------------------------------------------------------------------
     RF     =        40.00000
     NTAD   =        30
     ISERCH =         0
     LFRESH =         4
     ISWDIV =        22
     ITSTRT =         1
     ISOLAT =         1
     ZIMAX  =         2.00000
     ZDAMP  =         0.30000
     THKMIN =         0.50000
     THKPCT =         0.01000
     THKFUL =         0.99000
     THKRST =         0.02000
     IREST  =         1
     MESG1  =         0
     INT1PR =         1
     INTVAL =         1
     INTMAP =         1
     INTSAV =         0
     IPEQLB =         1
     INTCAL =         0
     INTSA2 =         0

INITIALIZE CELLS FOR DATA SET = DBOT     , LAYER =  1:

   SET ALL CELLS IN LAYER TO CONSTANT =  4.00000E+02

   PRINT OUT (FILE NO. 21) VALUES FOR ALL CELLS IN LAYER.

INITIALIZE CELLS FOR DATA SET = DBOT     , LAYER =  2:

   SET ALL CELLS IN LAYER TO CONSTANT =  3.00000E+02

INITIALIZE CELLS FOR DATA SET = DBOT     , LAYER =  3:

   SET ALL CELLS IN LAYER TO CONSTANT =  2.00000E+02

INITIALIZE CELLS FOR DATA SET = DBOT     , LAYER =  4:

   SET ALL CELLS IN LAYER TO CONSTANT =  1.00000E+02
```

```
INITIALIZE CELLS FOR DATA SET = DBOT    , LAYER =  5:

   SET ALL CELLS IN LAYER TO CONSTANT =  0.0

INITIALIZE CELLS FOR DATA SET = DBOT    , LAYER =  6:

   SET ALL CELLS IN LAYER TO CONSTANT = -1.00000E+02

*** START SEARCH FOR INTERFACE IN LAYER  1 ***

*** STARTING VALUES OF INTRUSION CODES FOR LAYER   1 ***

     0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
     1 = TRANSMISSIVITY > 0 AT START OF SIMULATION.
     3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0.
     4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0.
     5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0.

ROW
  1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

  2    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  4  4  0

  --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  4  4  0

 12    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

*** STARTING VALUES OF INTRUSION CODES FOR LAYER   2 ***

     0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
     1 = TRANSMISSIVITY > 0 AT START OF SIMULATION.
     3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0.
     4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0.
     5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0.

ROW
  1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

  2    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0

  3    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0

  4    0  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  4  0  0  0  0

  5    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0

  --- VALUES IN ROWS 6 TO 10 ARE THE SAME AS VALUES IN ROW 5 ---

 11    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0

 12    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

*** STARTING VALUES OF INTRUSION CODES FOR LAYER   3 ***

     0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
     1 = TRANSMISSIVITY > 0 AT START OF SIMULATION.
     3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0.
     4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0.
     5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0.

ROW
  1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

  2    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0
```

```
 --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0

 12   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

*** STARTING VALUES OF INTRUSION CODES FOR LAYER   4 ***

     0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
     1 = TRANSMISSIVITY > 0 AT START OF SIMULATION.
     3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0.
     4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0.
     5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0.

ROW
  1   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

  2   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0  0

 --- VALUES IN ROWS 3 TO 5 ARE THE SAME AS VALUES IN ROW 2 ---

  6   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0  0

  7   0  1  1  1  1  3  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0  0

  8   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0  0

 --- VALUES IN ROWS 9 AND 10 ARE THE SAME AS VALUES IN ROW 8 ---

 11   0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  4  0  0  0  0  0  0

 12   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

*** COMPUTATION OF ITERATION PARAMETERS FOR SIP EQUATION-SOLVING PROCEDURE ***

    NO. ITERATION PARAMETERS =   5

      0.0            7.8852643D-01   9.5527893D-01   9.9054268D-01   9.9800003D-01

DATA FROM RECORD 1 IN GROUP IV DATA: PARAMETERS FOR THE START OF A NEW STRESS PERIOD
-----------------------------------------------------------------------------------
     STRESS PERIOD NO. (KP)                                    =       1
     NUMBER OF TIME STEPS (NUMT)                               =       1
     TOTAL SIMULATION TIME (TMAX)                              =       1.0000 DAYS
     LENGTH OF FIRST TIME STEP (DELT)                          =      24.0000 HOURS
     MULTIPLIER FOR DELT (CDLT)                                =       1.0000
     CODE FOR READING NEW VALUES OF CONSTANT FLUX (IWEL)       =       0
     CODE FOR READING NEW VALUES OF RECHARGE (IRECH)           =       0
     CODE FOR READING NEW VALUES OF HEAD FOR RIVERS (IRIV)     =       0
     CODE FOR READING NEW VALUES OF CONSTANT HEAD (IHED)       =       0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =     1, TIME STEP =     1
------------------------------------------------------------------------------------------------------------------
     NO. ITERATIONS =  14

   HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
        -98.20   11, 16,  4          -62.42   2, 15,  5           -36.37   4, 14,  1           -29.67   6, 13,  1
        -12.93    4, 13,  1            4.63   2, 13,  6             4.76  11, 13,  6             4.99   2, 13,  6
          3.56    6, 11,  6            3.91   6, 13,  6            -0.45   5, 13,  6            -0.44   8, 13,  6
         -0.57   11, 13,  6           -0.46  10, 13,  6            -0.18  11, 13,  6

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    1 ***

     NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                        =     0
```

```
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                    =     0
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
     NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

     MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -9.995E+01  (I,J,K =   2, 21,  1)
     NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  40

     MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   0.0        (I,J,K =   0,  0,  0)
     NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =   0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
------------------------------------------------------------------------------------------------------------------
     NO. ITERATIONS =   0

   HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
        -0.08   11, 13,  5

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    2 ***

     NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                      =    39
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                    =     0
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
     NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

     MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -6.996E+01  (I,J,K =   2, 21,  1)
     NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  40

     MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   0.0        (I,J,K =   0,  0,  0)
     NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =   0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
------------------------------------------------------------------------------------------------------------------
     NO. ITERATIONS =   0

   HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
         0.07    2, 17,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    3 ***

     NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                      =    39
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                    =     0
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
     NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

     MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -4.897E+01  (I,J,K =   2, 21,  1)
     NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  38

     MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   0.0        (I,J,K =   0,  0,  0)
     NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =   0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
------------------------------------------------------------------------------------------------------------------
     NO. ITERATIONS =   0

   HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
         0.07    2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    4 ***

     NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                      =    39
```

```
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                          =     0
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
    NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0
    MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -3.427E+01  (I,J,K =   2, 21,  1)
    NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  37

    MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   5.657E-01  (I,J,K =  10, 18,  1)
    NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =    0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
------------------------------------------------------------------------------------------------------------------
    NO. ITERATIONS =   0

  HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
       0.05     2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    5 ***

    NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                             =    39
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                          =     0
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
    NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

    MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -2.398E+01  (I,J,K =   2, 21,  1)
    NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  36

    MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   1.497E+00  (I,J,K =  10, 18,  1)
    NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =    0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
------------------------------------------------------------------------------------------------------------------
    NO. ITERATIONS =   0

  HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
       0.04     2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    6 ***

    NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                             =    38
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                          =     0
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
    NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

    MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -1.678E+01  (I,J,K =   2, 21,  1)
    NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  35
    MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   1.444E+00  (I,J,K =  10, 18,  1)
    NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =    0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
------------------------------------------------------------------------------------------------------------------
    NO. ITERATIONS =   0

  HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
       0.03     2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    7 ***

    NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                             =    38
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                          =     0
    NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
```

```
NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -1.174E+01  (I,J,K =   2, 21,  1)
NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  34

MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   1.263E+00  (I,J,K =   9, 18,  1)
NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =   0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
-----------------------------------------------------------------------------------------------------------------
NO. ITERATIONS =   0

HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
     0.02    2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    8 ***

NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                        =    38
NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                     =     0
NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO    =     0
NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)     =     0

MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -8.209E+00  (I,J,K =   2, 21,  1)
NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  30

MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   1.228E+00  (I,J,K =   9, 18,  1)
NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =   0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
-----------------------------------------------------------------------------------------------------------------
NO. ITERATIONS =   0

HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
     0.01    2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.    9 ***

NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                        =    38
NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                     =     0
NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO    =     0
NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)     =     0

MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE     =  -5.737E+00  (I,J,K =   2, 21,  1)
NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX   =  30
MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE   =   1.078E+00  (I,J,K =   9, 18,  1)
NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =   0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =    1, TIME STEP =    1
-----------------------------------------------------------------------------------------------------------------
NO. ITERATIONS =   0

HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
     0.01    2, 18,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.   10 ***

NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                        =    38
NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO                     =     0
NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO    =     0
NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)     =     0
```

```
     MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE      =  -4.006E+00  (I,J,K =   2, 21,  1)
     NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX    =  30

     MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE    =   8.865E-01  (I,J,K =   9, 18,  1)
     NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =    0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =     1, TIME STEP =     1
--------------------------------------------------------------------------------------------------------------------
     NO. ITERATIONS =    0

   HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
          0.01   11, 19,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.   11 ***

     NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                  =    38
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO               =     0
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
     NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0

     MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE      =  -2.794E+00  (I,J,K =   2, 21,  1)
     NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX    =  20

     MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE    =   6.963E-01  (I,J,K =   9, 18,  1)
     NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =    0

MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SOLVING PROCEDURE.  STRESS PERIOD =     1, TIME STEP =     1
--------------------------------------------------------------------------------------------------------------------
     NO. ITERATIONS =    0

   HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)    HEAD CHANGE  CELL (I,J,K)
          0.00   11, 19,  1

*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING STRESS PERIOD NO.    1, TIME STEP NO.    1, TRIAL SOLUTION NO.   12 ***

     NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THE FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.

     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS ADJUSTED                  =    38
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO               =     0
     NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FROM ZERO TO NONZERO  =     0
     NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZERO (  1 PASSES)   =     0
     MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGENCE      =  -1.944E+00  (I,J,K =   2, 21,  1)
     NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX    =    0

     MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE    =   5.283E-01  (I,J,K =   9, 18,  1)
     NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX =    0

     ------------------------------------------------------------
     MASS BALANCE FOR STRESS PERIOD NO.    1, TIME STEP NO.    1
     ------------------------------------------------------------
        DURATION OF CURRENT TIME STEP      =      1.00000 DAYS
        DURATION OF CURRENT STRESS PERIOD  =      1.00000 DAYS
        TOTAL SIMULATION TIME              =      1.00000 DAYS

     MASS BALANCE FOR LAYER NO.    1, STRESS PERIOD NO.    1, TIME STEP NO.    1
     ---------------------------------------------------------------------------
        QUANTITY         RATE  IN       RATE OUT      CUMUL. VOL. IN      CUMUL. VOL. OUT
     --------------   ------------   ------------   ----------------   ----------------
     STORAGE                0.0          0.0                    0.                   0.
     RECHARGE               0.0          0.0                    0.                   0.
     CONST. FLUX            0.0          0.0                    0.                   0.
     PUMPING                0.0          0.0                    0.                   0.
     CONST. HEAD            0.0          0.0                    0.                   0.
     RIVERS                 0.0          0.001                  0.                  78.
```

DRAINS	0.0	0.0	0.	0.
ACROSS TOP	0.029	0.028	2505.	2425.
ACROSS BOT.	0.0	0.0	0.	0.
SUBTOTALS	0.029	0.029	2505.	2504.
IN - OUT	0.000		1.	
PCT. DIFF.	0.06		0.06	

MASS BALANCE FOR LAYER NO. 2, STRESS PERIOD NO. 1, TIME STEP NO. 1

QUANTITY	RATE IN	RATE OUT	CUMUL. VOL. IN	CUMUL. VOL. OUT
STORAGE	0.0	0.0	0.	0.
RECHARGE	0.0	0.0	0.	0.
CONST. FLUX	0.010	0.0	864.	0.
PUMPING	0.0	1.500	0.	129600.
CONST. HEAD	0.0	0.0	0.	0.
RIVERS	0.0	1.617	0.	139697.
DRAINS	0.0	0.0	0.	0.
ACROSS TOP	3.220	0.112	278230.	9677.
ACROSS BOT.	0.028	0.029	2425.	2505.
SUBTOTALS	3.258	3.258	281519.	281478.
IN - OUT	0.000		41.	
PCT. DIFF.	0.01		0.01	

MASS BALANCE FOR LAYER NO. 3, STRESS PERIOD NO. 1, TIME STEP NO. 1

QUANTITY	RATE IN	RATE OUT	CUMUL. VOL. IN	CUMUL. VOL. OUT
STORAGE	0.0	0.0	0.	0.
RECHARGE	0.0	0.0	0.	0.
CONST. FLUX	0.0	0.0	0.	0.
PUMPING	0.0	0.0	0.	0.
CONST. HEAD	0.0	0.0	0.	0.
RIVERS	0.0	0.049	0.	4211.
DRAINS	0.0	0.0	0.	0.
ACROSS TOP	3.251	0.094	280895.	8125.
ACROSS BOT.	0.112	3.220	9677.	278230.
SUBTOTALS	3.363	3.363	290572.	290566.
IN - OUT	0.000		5.	
PCT. DIFF.	0.00		0.00	

MASS BALANCE FOR LAYER NO. 4, STRESS PERIOD NO. 1, TIME STEP NO. 1

QUANTITY	RATE IN	RATE OUT	CUMUL. VOL. IN	CUMUL. VOL. OUT
STORAGE	0.0	0.0	0.	0.
RECHARGE	0.0	0.0	0.	0.
CONST. FLUX	0.010	0.0	864.	0.
PUMPING	0.0	1.000	0.	86400.
CONST. HEAD	0.0	0.0	0.	0.
RIVERS	0.0	4.720	0.	407789.
DRAINS	0.0	0.0	0.	0.
ACROSS TOP	8.878	0.010	767063.	831.
ACROSS BOT.	0.094	3.251	8125.	280895.
SUBTOTALS	8.982	8.981	776052.	775915.
IN - OUT	0.002		137.	
PCT. DIFF.	0.02		0.02	

MASS BALANCE FOR LAYER NO. 5, STRESS PERIOD NO. 1, TIME STEP NO. 1

QUANTITY	RATE IN	RATE OUT	CUMUL. VOL. IN	CUMUL. VOL. OUT
STORAGE	0.0	0.0	0.	0.
RECHARGE	0.006	0.0	518.	0.
CONST. FLUX	0.0	0.0	0.	0.
PUMPING	0.0	0.0	0.	0.
CONST. HEAD	0.0	0.0	0.	0.
RIVERS	0.0	0.0	0.	0.
DRAINS	0.0	0.135	0.	11674.
ACROSS TOP	8.998	0.0	777402.	0.
ACROSS BOT.	0.010	8.878	831.	767063.
SUBTOTALS	9.013	9.013	778752.	778737.
IN - OUT		0.000		15.
PCT. DIFF.		0.00		0.00

MASS BALANCE FOR LAYER NO. 6, STRESS PERIOD NO. 1, TIME STEP NO. 1

QUANTITY	RATE IN	RATE OUT	CUMUL. VOL. IN	CUMUL. VOL. OUT
STORAGE	0.0	0.0	0.	0.
RECHARGE	10.280	0.0	888191.	0.
CONST. FLUX	0.010	0.0	864.	0.
PUMPING	0.0	0.0	0.	0.
CONST. HEAD	0.0	0.0	0.	0.
RIVERS	0.0	0.0	0.	0.
DRAINS	0.0	1.288	0.	111293.
ACROSS TOP	0.0	0.0	0.	0.
ACROSS BOT.	0.0	8.998	0.	777402.
SUBTOTALS	10.290	10.286	889055.	888695.
IN - OUT		0.004		360.
PCT. DIFF.		0.04		0.04

MASS BALANCE FOR ENTIRE MODEL, STRESS PERIOD NO. 1, TIME STEP NO. 1

QUANTITY	RATE IN	RATE OUT	CUMUL. VOL. IN	CUMUL. VOL. OUT
STORAGE	0.0	0.0	0.	0.
RECHARGE	10.286	0.0	888709.	0.
CONST. FLUX	0.030	0.0	2592.	0.
PUMPING	0.0	2.500	0.	216000.
CONST. HEAD	0.0	0.0	0.	0.
RIVERS	0.0	6.386	0.	551775.
DRAINS	0.0	1.423	0.	122967.
SUBTOTALS	10.316	10.310	891301.	890742.
IN - OUT		0.006		560.
PCT. DIFF.		0.06		0.06

COMPUTED FLUX ACROSS TOPS OF MODEL LAYERS, STRESS PERIOD NO. 1, TIME STEP NO. 1

LAYER	UPWARD FLUX	DOWNWARD FLUX
1	2.807275E-02	2.899561E-02
2	1.119970E-01	3.220253E+00
3	9.404165E-02	3.251101E+00
4	9.621818E-03	8.878042E+00
5	0.0	8.997712E+00

COMPUTED FLUX FOR STRESS PERIOD 1, TIME STEP 1, SIMULATION TIME OF 86400.0

(NEGATIVE VALUES INDICATE FLUX OUT OF AQUIFER)

FLUX BETWEEN AQUIFER AND CONSTANT-HEAD CELLS

CONSTANT-HEAD CELL (K, I, J)	AQUIFER CELL (K, I, J)	FLUX (FT**3/SEC)

FLUX BETWEEN AQUIFER AND RIVER CELLS

AQUIFER CELL (K, I, J)	FLUX (FT**3/SEC)
1, 2, 19	-3.9686E-05
1, 2, 20	-1.6833E-06
1, 2, 21	-1.4103E-07
1, 3, 19	-4.1220E-05
1, 3, 20	-1.7430E-06
1, 3, 21	-1.4565E-07
1, 4, 19	-4.6252E-05
1, 4, 20	-1.9235E-06
1, 4, 21	-1.5836E-07
1, 5, 19	-5.6889E-05
1, 5, 20	-2.2846E-06
1, 5, 21	-1.8224E-07
1, 6, 19	-7.2404E-05
1, 6, 20	-2.7960E-06
1, 6, 21	-2.1457E-07
1, 7, 19	-9.0108E-05
1, 7, 20	-3.3694E-06
1, 7, 21	-2.4945E-07
1, 8, 19	-1.0704E-04
1, 8, 20	-3.9119E-06
1, 8, 21	-2.8147E-07
1, 9, 19	-1.2103E-04
1, 9, 20	-4.3574E-06
1, 9, 21	-3.0722E-07
1, 10, 19	-1.3093E-04
1, 10, 20	-4.6718E-06
1, 10, 21	-3.2518E-07
1, 11, 19	-1.3608E-04
1, 11, 20	-4.8418E-06
1, 11, 21	-3.3596E-07
2, 2, 18	-1.3448E-01
2, 3, 18	-1.3559E-01
2, 4, 18	-1.3931E-01
2, 5, 18	-1.4690E-01
2, 6, 18	-1.5693E-01
2, 7, 18	-1.6721E-01
2, 8, 18	-1.7616E-01
2, 9, 18	-1.8300E-01
2, 10, 18	-1.8753E-01
2, 11, 18	-1.8976E-01
3, 2, 17	-4.2378E-03
3, 3, 17	-4.2558E-03
3, 4, 17	-4.3362E-03
3, 5, 17	-4.5188E-03
3, 6, 17	-4.7625E-03
3, 7, 17	-5.0090E-03
3, 8, 17	-5.2203E-03
3, 9, 17	-5.3803E-03
3, 10, 17	-5.4856E-03
3, 11, 17	-5.5374E-03
4, 2, 16	-4.6192E-01
4, 3, 16	-4.6227E-01
4, 4, 16	-4.6354E-01
4, 5, 16	-4.6628E-01
4, 6, 16	-4.6995E-01
4, 7, 16	-4.7379E-01
4, 8, 16	-4.7724E-01

FLUX INTO DRAINS

AQUIFER CELL (K, I, J)	FLUX (FT**3/SEC)

```
                                        4,  9, 16    -4.8000E-01
                                        4, 10, 16    -4.8191E-01
                                        4, 11, 16    -4.8289E-01
                                                                     5,  2, 15    -1.3320E-02
                                                                     5,  3, 15    -1.3326E-02
                                                                     5,  4, 15    -1.3349E-02
                                                                     5,  5, 15    -1.3401E-02
                                                                     5,  6, 15    -1.3472E-02
                                                                     5,  7, 15    -1.3546E-02
                                                                     5,  8, 15    -1.3613E-02
                                                                     5,  9, 15    -1.3666E-02
                                                                     5, 10, 15    -1.3704E-02
                                                                     5, 11, 15    -1.3723E-02
                                                                     6,  2, 14    -1.2675E-01
                                                                     6,  3, 14    -1.2675E-01
                                                                     6,  4, 14    -1.2695E-01
                                                                     6,  5, 14    -1.2750E-01
                                                                     6,  6, 14    -1.2827E-01
                                                                     6,  7, 14    -1.2910E-01
                                                                     6,  8, 14    -1.2989E-01
                                                                     6,  9, 14    -1.3056E-01
                                                                     6, 10, 14    -1.3104E-01
                                                                     6, 11, 14    -1.3129E-01

*** INTRUSION CODES AT END OF TIME STEP FOR LAYER   1 ***

    NO. CELLS WITH TRANSMISSIVITY SET TO ZERO SINCE START OF SIMULATION =      0
    NO. CELLS WITH TRANSMISSIVITY ADJUSTED SINCE START OF SIMULATION    =     38

     0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
     1 = TRANSMISSIVITY > 0 AT START OF SIMULATION AND NO INTRUSION.
    -1 = TRANSMISSIVITY SET TO ZERO BECAUSE ALL OF CELL IS INTRUDED OR BECAUSE CELL IS ISOLATED.
    -2 = TRANSMISSIVITY ADJUSTED BECAUSE PART OF CELL IS INTRUDED.
     3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
    -3 = WELL CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
     4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
    -4 = LEAKY-OCEAN CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
     5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
    -5 = LEAKY-RIVER CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.

ROW
  1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

  2    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -2 -4 -4 -4  0

  --- VALUES IN ROWS 3 TO 8 ARE THE SAME AS VALUES IN ROW 2 ---

  9    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -2 -4 -4 -4  0

 10    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -4 -4 -4  0

 11    0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -4 -4 -4  0

 12    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

*** INTRUSION CODES AT END OF TIME STEP FOR LAYER   2 ***

    NO. CELLS WITH TRANSMISSIVITY SET TO ZERO SINCE START OF SIMULATION =      0
    NO. CELLS WITH TRANSMISSIVITY ADJUSTED SINCE START OF SIMULATION    =      0

     0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
     1 = TRANSMISSIVITY > 0 AT START OF SIMULATION AND NO INTRUSION.
    -1 = TRANSMISSIVITY SET TO ZERO BECAUSE ALL OF CELL IS INTRUDED OR BECAUSE CELL IS ISOLATED.
    -2 = TRANSMISSIVITY ADJUSTED BECAUSE PART OF CELL IS INTRUDED.
     3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
    -3 = WELL CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
```

```
}

      4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -4 = LEAKY-OCEAN CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
      5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -5 = LEAKY-RIVER CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.

ROW
  1    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

  2    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 D 0 0 0

  3    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0

  4    0 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 4 0 0 0 0

  5    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0

  ---VALUES IN ROWS 6 TO 10 ARE THE SAME AS VALUES IN ROW 5 ---

 11    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0

 12    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

*** INTRUSION CODES AT END OF TIME STEP FOR LAYER   3 ***

     NO. CELLS WITH TRANSMISSIVITY SET TO ZERO SINCE START OF SIMULATION =      0
     NO. CELLS WITH TRANSMISSIVITY ADJUSTED SINCE START OF SIMULATION    =      0

      0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
      1 = TRANSMISSIVITY > 0 AT START OF SIMULATION AND NO INTRUSION.
     -1 = TRANSMISSIVITY SET TO ZERO BECAUSE ALL OF CELL IS INTRUDED OR BECAUSE CELL IS ISOLATED.
     -2 = TRANSMISSIVITY ADJUSTED BECAUSE PART OF CELL IS INTRUDED.
      3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -3 = WELL CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
      4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -4 = LEAKY-OCEAN CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
      5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -5 = LEAKY-RIVER CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.

ROW
  1    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0 0 0 0

  2    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0

  --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0

 12    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

*** INTRUSION CODES AT END OF TIME STEP FOR LAYER   4 ***

     NO. CELLS WITH TRANSMISSIVITY SET TO ZERO SINCE START OF SIMULATION =      0
     NO. CELLS WITH TRANSMISSIVITY ADJUSTED SINCE START OF SIMULATION    =      0

      0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.
      1 = TRANSMISSIVITY > D AT START OF SIMULATION AND NO INTRUSION.
     -1 = TRANSMISSIVITY SET TO ZERO BECAUSE ALL OF CELL IS INTRUDED OR BECAUSE CELL IS ISOLATED.
     -2 = TRANSMISSIVITY ADJUSTED BECAUSE PART OF CELL IS INTRUDED.
      3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -3 = WELL CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
      4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -4 = LEAKY-OCEAN CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
      5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0 AND NO INTRUSION.
     -5 = LEAKY-RIVER CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUSTED.
```

```
ROW
  1    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

  2    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0 0

  --- VALUES IN ROWS 3 TO 5 ARE THE SAME AS VALUES IN ROW 2 ---

  6    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0 0

  7    0 1 1 1 1 3 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0 0

  8    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0 0

  -- VALUES IN ROWS 9 TO 10 ARE THE SAME AS VALUES IN ROW 8 --

 11    0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 0 0 0 0 0 0

 12    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

*** SUMMARY OF ALL CELLS WITH ADJUSTED TRANSMISSIVITIES AT END OF TIME STEP ***

LAYER	ROW	COL.	DEPTH TO SALTWATER	ORIGINAL THICKNESS	ADJUSTED THICKNESS
1	2	18	355.67	100.00	55.67
1	2	19	310.57	100.00	10.57
1	2	20	303.11	100.00	3.11
1	2	21	302.11	100.00	2.11
1	3	18	356.65	100.00	56.65
1	3	19	310.75	100.00	10.75
1	3	20	303.14	100.00	3.14
1	3	21	302.12	100.00	2.12
1	4	18	360.01	100.00	60.01
1	4	19	311.34	100.00	11.34
1	4	20	303.23	100.00	3.23
1	4	21	302.13	100.00	2.13
1	5	18	366.85	100.00	66.85
1	5	19	312.49	100.00	12.49
1	5	20	303.40	100.00	3.40
1	5	21	302.15	100.00	2.15
1	6	18	375.80	100.00	75.80
1	6	19	313.98	100.00	13.98
1	6	20	303.62	100.00	3.62
1	6	21	302.18	100.00	2.18
1	7	18	384.88	100.00	84.88
1	7	19	315.51	100.00	15.51
1	7	20	303.85	100.00	3.85
1	7	21	302.21	100.00	2.21
1	8	18	392.71	100.00	92.71
1	8	19	316.84	100.00	16.84
1	8	20	304.05	100.00	4.05
1	8	21	302.23	100.00	2.23
1	9	18	398.69	100.00	98.69
1	9	19	317.86	100.00	17.86
1	9	20	304.21	100.00	4.21
1	9	21	302.25	100.00	2.25
1	10	19	318.54	100.00	18.54
1	10	20	304.31	100.00	4.31
1	10	21	302.27	100.00	2.27
1	11	19	318.89	100.00	18.89
1	11	20	304.37	100.00	4.37
1	11	21	302.28	100.00	2.28

```
*** GREATEST DEPTHS TO FRESHWATER-SALTWATER INTERFACE ***

     VALUES > 0 ARE DEPTHS TO INTERFACE OR BOTTOM OF MODEL CELL
     -77 = TOTAL INTRUSION IN A VERTICAL COLUMN OF CELLS
     -99 = NO-FLOW BOUNDARY

ROW
  1 -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.
    -99.  -99.

  2 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  356.  311.  303.
    302.  -99.

  3 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  357.  311.  303.
    302.  -99.

  4 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  360.  311.  303.
    302.  -99.

  5 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  367.  312.  303.
    302.  -99.

  6 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  376.  314.  304.
    302.  -99.

  7 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  385.  316.  304.
    302.  -99.

  8 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  393.  317.  304.
    302.  -99.

  9 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  399.  318.  304.
    302.  -99.

 10 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  319.  304.
    302.  -99.

 11 -99.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  400.  319.  304.
    302.  -99.

 12 -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.  -99.
    -99.  -99.

*** INTRUSION CODES FOR CELLS USED TO DETERMINE GREATEST DEPTHS TO FRESHWATER-SALTWATER INTERFACE ***-

ROW
  1   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.

  2   0.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.   -2.   -4.   -4.
     -4.    0.

 -- VALUES IN ROWS 3 TO 8 ARE THE SAME AS VALUES IN ROW 2 --

  9   0.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.   -2.   -4.   -4.
     -4.    0.

 10   0.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.   -4.   -4.
     -4.    0.

 11   0.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.    1.   -4.   -4.
     -4.    0.

 12   0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
      0.    0.
```

```
*** DIFFERENCE BETWEEN EQUILIBRIUM AND COMPUTED INTERFACE POSITIONS FOR LAYER  1 ***

ROW
  1   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

  2   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  -1.  -2.
     -2.   0.

 -- VALUES IN ROWS 3 TO 7 ARE THE SAME AS VALUES IN ROW 2 --

  8   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  -1.  -2.
     -2.   0.

  9   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   1.  -1.  -2.
     -2.   0.

 10   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  -1.  -2.
     -2.   0.

 11   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  -1.  -2.
     -2.   0.

 12   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

*** DIFFERENCE BETWEEN EQUILIBRIUM AND COMPUTED INTERFACE POSITIONS FOR LAYER  2 ***

ROW
  1   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

*** DIFFERENCE BETWEEN EQUILIBRIUM AND COMPUTED INTERFACE POSITIONS FOR LAYER  3 ***

ROW
  1   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

*** DIFFERENCE BETWEEN EQUILIBRIUM AND COMPUTED INTERFACE POSITIONS FOR LAYER  4 ***

ROW
  1   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      0.   0.
```

VALUES ALONG VERTICAL SECTION NO. 1 FOR PARAMETER = CALCHEAD PAGE NO.

......................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 8)	(4, 9)	(4, 10)	(4, 11)
6	-99.0000	230.4079	229.5941	228.0162	225.6969	222.6685	218.9597	214.5701	209.4994	203.7735	197.4678
5	-99.0000	193.1875	192.0232	189.7610	186.4801	182.3373	177.5047	171.7501	164.8535	156.6715	147.1077
4	-99.0000	156.1323	154.6138	151.6695	147.4299	142.1747	136.2196	129.1004	120.3780	109.7392	96.9230
3	-99.0000	138.7077	137.2318	134.3468	130.1209	124.6899	118.1865	110.3046	100.6831	88.8536	73.8465
2	-99.0000	121.4736	120.0364	117.2135	113.0061	107.4044	100.3583	91.7205	81.2083	68.1913	50.8315
1	-99.0000	121.2975	119.8656	117.0439	112.8367	107.2334	100.1820	91.5371	81.0200	68.0349	51.1548

VALUES ALONG VERTICAL SECTION NO. 1 FOR PARAMETER = CALCHEAD PAGE NO.

......................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(4, 12)	(4, 13)	(4, 14)	(4, 15)	(4, 16)	(4, 17)	(4, 18)	(4, 19)	(4, 20)	(4, 21)	(4, 22)
6	190.7422	183.9120	102.7010	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
5	136.0148	121.5296	72.9209	25.1862	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
4	81.6828	63.8844	44.1504	25.8592	8.7461	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
3	53.8340	47.0725	36.0063	24.7425	14.4003	10.6814	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
2	22.5034	30.3498	27.9811	23.4954	18.5425	13.4471	7.9641	-99.0000	-99.0000	-99.0000	-99.0000
1	28.6248	30.7337	27.9603	23.4964	18.6267	13.8108	9.0028	7.7549	7.5372	7.5047	-99.0000

VALUES ALONG VERTICAL SECTION NO. 2 FOR PARAMETER = CALCHEAD PAGE NO.

......................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)	(7, 8)	(7, 9)	(7, 10)	(7, 11)
6	-99.0000	230.2086	229.3628	227.7186	225.3013	222.1848	218.5975	214.3524	209.4034	203.7685	197.5202
5	-99.0000	192.5163	191.0495	187.9631	182.6565	173.9041	173.8481	170.3037	164.4463	156.8313	147.6135
4	-99.0000	154.9934	152.9034	148.3761	140.0754	123.3307	129.1659	126.4327	119.6721	110.0818	97.9147
3	-99.0000	138.2653	136.5186	132.8787	126.7477	116.9068	115.5103	110.3542	102.6232	92.9264	81.5579
2	-99.0000	121.7300	120.3238	117.5719	113.5048	108.2101	101.9499	94.4871	85.7961	76.0029	65.4474
1	-99.0000	121.5609	120.1630	117.4203	113.3720	108.1079	101.8140	94.3303	85.6301	75.8363	65.2902

VALUES ALONG VERTICAL SECTION NO. 2 FOR PARAMETER = CALCHEAD PAGE NO.

......................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(7, 12)	(7, 13)	(7, 14)	(7, 15)	(7, 16)	(7, 17)	(7, 18)	(7, 19)	(7, 20)	(7, 21)	(7, 22)
6	190.8117	183.9451	102.7469	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
5	136.6985	122.1185	73.3677	25.5583	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
4	83.0410	65.0552	45.0013	26.3888	8.9394	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
3	68.7791	54.7906	40.3948	27.4774	16.0922	11.9509	-99.0000	-99.0000	-99.0000	-99.0000	-99.0000
2	54.7822	44.7988	35.9162	28.4277	21.6461	15.2163	8.5577	-99.0000	-99.0000	-99.0000	-99.0000
1	54.6492	44.7029	35.8735	28.4405	21.7267	15.4812	9.6312	7.8631	7.5547	7.5071	-99.0000

VALUES ALONG VERTICAL SECTION NO. 1 FOR PARAMETER = INTRUCOD PAGE NO.

......................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 8)	(4, 9)	(4, 10)	(4, 11)
6	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

VALUES ALONG VERTICAL SECTION NO. 1 FOR PARAMETER = INTRUCOD PAGE NO.

......................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(4, 12)	(4, 13)	(4, 14)	(4, 15)	(4, 16)	(4, 17)	(4, 18)	(4, 19)	(4, 20)	(4, 21)	(4, 22)
6	1.0000	1.0000	1.0000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.0000	1.0000	1.0000	1.0000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1.0000	1.0000	1.0000	1.0000	4.0000	0.0	0.0	0.0	0.0	0.0	0.0
3	1.0000	1.0000	1.0000	1.0000	1.0000	4.0000	0.0	0.0	0.0	0.0	0.0
2	3.0000	1.0000	1.0000	1.0000	1.0000	1.0000	4.0000	0.0	0.0	0.0	0.0
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	-2.0000	-4.0000	-4.0000	-4.0000	0.0

VALUES ALONG VERTICAL SECTION NO. 2 FOR PARAMETER = INTRUCOD PAGE NO.

....................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)	(7, 8)	(7, 9)	(7, 10)	(7, 11)
6	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	0.0	1.0000	1.0000	1.0000	1.0000	3.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

VALUES ALONG VERTICAL SECTION NO. 2 FOR PARAMETER = INTRUCOD PAGE NO.

....................PARAMETER VALUES FOR SPECIFIED ROW AND COLUMN COORDINATES (I,J) IN EACH LAYER......................

LAYER	(7, 12)	(7, 13)	(7, 14)	(7, 15)	(7, 16)	(7, 17)	(7, 18)	(7, 19)	(7, 20)	(7, 21)	(7, 22)
6	1.0000	1.0000	1.0000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.0000	1.0000	1.0000	1.0000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1.0000	1.0000	1.0000	1.0000	4.0000	0.0	0.0	0.0	0.0	0.0	0.0
3	1.0000	1.0000	1.0000	1.0000	1.0000	4.0000	0.0	0.0	0.0	0.0	0.0
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	4.0000	0.0	0.0	0.0	0.0
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	-2.0000	-4.0000	-4.0000	-4.0000	0.0

*** END OF JOB ***

COMPUTED HEAD FOR LAYER 2. STRESS PERIOD = 1, TIME STEP = 1, SIMULATION TIME = 86400.00

ROW											
1	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00
	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
2	-99.00	121.45	120.01	117.17	112.92	107.24	100.06	91.32	80.99	69.16	56.38	44.27
	35.88	29.19	23.53	18.27	13.19	7.86	-99.00	-99.00	-99.00	-99.00		
3	-99.00	121.46	120.02	117.18	112.95	107.29	100.14	91.40	80.92	68.50	54.12	39.06
	33.62	28.44	23.33	18.27	13.24	7.88	-99.00	-99.00	-99.00	-99.00		
4	-99.00	121.47	120.04	117.21	113.01	107.40	100.36	91.72	81.21	68.19	50.83	22.50
	30.35	27.98	23.50	18.54	13.45	7.96	-99.00	-99.00	-99.00	-99.00		
5	-99.00	121.51	120.08	117.27	113.10	107.60	100.76	92.44	82.46	70.58	56.65	41.78
	36.12	30.47	24.89	19.39	13.93	8.13	-99.00	-99.00	-99.00	-99.00		
6	-99.00	121.60	120.17	117.38	113.26	107.86	101.30	93.43	84.14	73.49	61.80	50.21
	41.21	33.43	26.73	20.53	14.57	8.34	-99.00	-99.00	-99.00	-99.00		
7	-99.00	121.73	120.32	117.57	113.50	108.21	101.95	94.49	85.80	76.00	65.45	54.78
	44.80	35.92	28.43	21.65	15.22	8.56	-99.00	-99.00	-99.00	-99.00		
8	-99.00	121.91	120.54	117.87	113.93	108.84	102.73	95.52	87.22	77.94	67.93	57.56
	47.21	37.76	29.78	22.57	15.77	8.75	-99.00	-99.00	-99.00	-99.00		
9	-99.00	122.11	120.78	118.19	114.38	109.44	103.44	96.39	88.32	79.32	69.57	59.31
	48.79	39.03	30.74	23.26	16.19	8.89	-99.00	-99.00	-99.00	-99.00		
10	-99.00	122.28	120.98	118.44	114.72	109.88	103.96	97.01	89.07	80.21	70.58	60.35
	49.75	39.82	31.36	23.70	16.46	8.99	-99.00	-99.00	-99.00	-99.00		
11	-99.00	122.37	121.09	118.58	114.91	110.11	104.24	97.33	89.44	80.65	71.06	60.84
	50.20	40.20	31.66	23.92	16.60	9.04	-99.00	-99.00	-99.00	-99.00		
12	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00
	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		

COMPUTED HEAD FOR LAYER 4. STRESS PERIOD = 1, TIME STEP = 1, SIMULATION TIME = 86400.00

ROW												
1	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00
	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
2	-99.00	156.74	155.37	152.73	148.86	143.80	137.55	129.96	120.84	109.94	97.03	81.81
	63.86	44.07	25.79	8.72	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
3	-99.00	156.52	155.11	152.38	148.41	143.31	137.13	129.68	120.67	109.85	96.95	81.73
	63.83	44.07	25.80	8.72	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
4	-99.00	156.13	154.61	151.67	147.43	142.17	136.22	129.10	120.38	109.74	96.92	81.68
	63.88	44.15	25.86	8.75	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
5	-99.00	155.64	153.94	150.59	145.73	139.93	134.61	128.22	120.02	109.73	97.15	82.08
	64.21	44.38	26.00	8.80	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
6	-99.00	155.19	153.26	149.29	143.10	135.21	132.08	127.14	119.70	109.82	97.50	82.56
	64.63	44.69	26.19	8.87	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
7	-99.00	154.99	152.90	148.38	140.08	123.33	129.17	126.43	119.67	110.08	97.91	83.04
	65.06	45.00	26.39	8.94	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
8	-99.00	155.17	153.25	149.31	143.16	135.35	132.34	127.53	120.26	110.55	98.37	83.48
	65.43	45.28	26.57	9.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
9	-99.00	155.57	153.89	150.59	145.81	140.16	135.05	128.93	121.05	111.09	98.81	83.86
	65.74	45.51	26.71	9.06	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
10	-99.00	155.96	154.46	151.57	147.43	142.37	136.71	129.97	121.70	111.54	99.15	84.13
	65.96	45.67	26.81	9.09	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
11	-99.00	156.20	154.78	152.07	148.18	143.28	137.47	130.51	122.06	111.78	99.33	84.28
	66.07	45.75	26.86	9.11	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
12	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00
	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		

COMPUTED HEAD FOR LAYER 6. STRESS PERIOD = 1, TIME STEP = 1, SIMULATION TIME = 86400.00

ROW												
1	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00
	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
2	-99.00	230.52	229.72	228.17	225.87	222.85	219.12	214.69	209.58	203.82	197.49	190.75
	183.92	102.70	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
3	-99.00	230.48	229.68	228.11	225.81	222.79	219.07	214.65	209.55	203.80	197.48	190.74
	183.91	102.70	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
4	-99.00	230.41	229.59	228.02	225.70	222.67	218.96	214.57	209.50	203.77	197.47	190.74
	183.91	102.70	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
5	-99.00	230.32	229.50	227.89	225.55	222.50	218.82	214.47	209.44	203.75	197.47	190.75
	183.92	102.71	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
6	-99.00	230.25	229.41	227.78	225.39	222.32	218.67	214.38	209.40	203.74	197.48	190.78
	183.93	102.73	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
7	-99.00	230.21	229.36	227.72	225.30	222.18	218.60	214.35	209.40	203.77	197.52	190.81
	183.95	102.75	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		
8	-99.00	230.22	229.38	227.75	225.38	222.32	218.69	214.42	209.46	203.83	197.57	190.86
	183.97	102.76	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00	-99.00		

```
  9   -99.00   230.26   229.43   227.84   225.51   222.49   218.84   214.54   209.55   203.90   197.64   190.91
      184.00   102.78   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00

 10   -99.00   230.30   229.49   227.93   225.63   222.64   218.97   214.64   209.64   203.96   197.69   190.95
      184.02   102.79   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00

 11   -99.00   230.33   229.53   227.98   225.70   222.71   219.05   214.71   209.69   204.00   197.72   190.97
      184.04   102.79   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00

 12   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00
      -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00   -99.00

                   VALUES FOR DATA SET = STRT    , LAYER =  1, SIMULATION TIME =       0.0
                   --------------------------------------------------------------------
ROW
  1  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.
     110.  110.  110.  110.

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.  110.
     110.  110.  110.  110.

                   VALUES FOR DATA SET = S       , LAYER =  1, SIMULATION TIME =       0.0
                   --------------------------------------------------------------------
ROW
  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

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12        0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
           0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
           0.0         0.0

                   VALUES FOR DATA SET = T       , LAYER =  2, SIMULATION TIME =       0.0
                   --------------------------------------------------------------------
ROW
  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

  2 0.0        0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000
    0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.0        0.0
    0.0        0.0

 --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11 0.0        0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000
    0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.02000000 0.0        0.0
    0.0        0.0

 12 0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0

                   VALUES FOR DATA SET = TXTY    , LAYER =  1, SIMULATION TIME =       0.0
                   --------------------------------------------------------------------
ROW
  1 1.0000E+00 1.0000E+00 1.0000E-04

                   VALUES FOR DATA SET = PERM    , LAYER =  1, SIMULATION TIME =       0.0
                   --------------------------------------------------------------------
```

```
ROW
  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

  2 0.0        0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000
    0.00020000 0.00020000 0.00020000 0.00020000 0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0

 --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11 0.0        0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000 0.00020000
    0.00020000 0.00020000 0.00020000 0.00020000 0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0

 12 0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0

                VALUES FOR DATA SET = BOT     , LAYER =  1, SIMULATION TIME =       0.0
                ----------------------------------------------------------------------
ROW
  1      100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00
         100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00
         100.00     100.00

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12      100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00
         100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00     100.00
         100.00     100.00

                VALUES FOR DATA SET = QRE     , LAYER =  1, SIMULATION TIME =       0.0
                ----------------------------------------------------------------------
ROW
  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

  2 0.0        2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08
    2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-10 0.0        0.0        0.0        0.0        0.0
    0.0        0.0

  --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11 0.0        2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08
    2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-08 2.0000E-10 0.0        0.0        0.0        0.0        0.0
    0.0        0.0

 12 0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
    0.0        0.0

                VALUES FOR DATA SET = DELX    , LAYER =  1, SIMULATION TIME =       0.0
                ----------------------------------------------------------------------
ROW
  1     2000.00    2000.00    2000.00    2000.00    2000.00    2000.00    2000.00    2000.00    2000.00    2000.00
        2000.00    2000.00    2000.00    1700.00    1500.00    1500.00    1500.00    1700.00    2000.00    2000.00
        2000.00    2000.00

                VALUES FOR DATA SET = WELL    , LAYER =  2, SIMULATION TIME =       0.0
                ----------------------------------------------------------------------
```

ROW										
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								
2	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	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	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	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	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	-1.500	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								
5	0.0	0.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								

--- VALUES IN ROWS 6 TO 11 ARE THE SAME AS VALUES IN ROW 5 ---

12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								

VALUES FOR DATA SET = WELL , LAYER = 4, SIMULATION TIME = 0.0

ROW										
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								
2	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								

--- VALUES IN ROWS 3 TO 6 ARE THE SAME AS VALUES IN ROW 2 ---

7	0.0	0.001	0.0	0.0	0.0	-1.00	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								
8	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								

--- VALUES IN ROWS 9 TO 11 ARE THE SAME AS VALUES IN ROW 8 ---

12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								

VALUES FOR DATA SET = WELL , LAYER = 6, SIMULATION TIME = 0.0

ROW										
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								
2	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
 --- VALUES IN ROWS 3 TO 11 ARE THE SAME AS VALUES IN ROW 2 ---

 12       0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0

                VALUES FOR DATA SET = IDN     , LAYER =  5, SIMULATION TIME =       0.0
                -------------------------------------------------------------------------
ROW
  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

  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        1.00       0.0        0.0        0.0        0.0        0.0
          0.0        0.0

 --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11       0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0        0.0        0.0        1.00       0.0        0.0        0.0        0.0        0.0
          0.0        0.0

 12       0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0

                VALUES FOR DATA SET = IDN     , LAYER =  6, SIMULATION TIME =       0.0
                -------------------------------------------------------------------------
ROW
  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

  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        1.00       0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0

 --- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

 11       0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0        0.0        1.00       0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0

 12       0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
          0.0        0.0

                VALUES FOR DATA SET = DRCF    , LAYER =  5, SIMULATION TIME =       0.0
                -------------------------------------------------------------------------
ROW
  1 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04
    5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04
    5.3000E-04 5.3000E-04

 --- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

 12 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04
    5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04 5.3000E-04
    5.3000E-04 5.3000E-04

                VALUES FOR DATA SET = DREL    , LAYER =  6, SIMULATION TIME =       0.0
                -------------------------------------------------------------------------
```

ROW										
1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	100.00	100.00								

--- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

12	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	100.00	100.00								

VALUES FOR DATA SET = IDR , LAYER = 1, SIMULATION TIME = 0.0

ROW										
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								
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	2.00	2.00
	2.00	0.0								

--- VALUES IN ROWS 3 TO 10 ARE THE SAME AS VALUES IN ROW 2 ---

11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.00	2.00
	2.00	0.0								
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0								

VALUES FOR DATA SET = RC , LAYER = 1, SIMULATION TIME = 0.0

ROW										
1	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03
	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03
	1.6000E-03	1.6000E-03								

--- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

12	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03
	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03	1.6000E-03
	1.6000E-03	1.6000E-03								

VALUES FOR DATA SET = RB , LAYER = 1, SIMULATION TIME = 0.0

ROW										
1	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00
	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00
	-300.00	-300.00								

--- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

12	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00
	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00	-300.00
	-300.00	-300.00								

VALUES FOR DATA SET = RH , LAYER = 1, SIMULATION TIME = 0.0

ROW										
1	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50
	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50
	7.50	7.50								

--- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

12	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50
	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50
	7.50	7.50								

VALUES FOR DATA SET = DBOT , LAYER = 1, SIMULATION TIME = 0.0

ROW										
1	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00
	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00
	400.00	400.00								

--- VALUES IN ROWS 2 TO 11 ARE THE SAME AS VALUES IN ROW 1 ---

12	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00
	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00
	400.00	400.00								

page 89 follows

A P P E N D I X

```
C...PROGRAM SSIM3D...                                                   AA00010*
C     ------------------------------------------------------------------AA00020
C     FINITE-DIFFERENCE MODEL FOR SIMULATION OF GROUND-WATER FLOW IN    AA00030
C     THREE DIMENSIONS, SEPTEMBER, 1975 BY P.C. TRESCOTT, U. S. G. S.   AA00040
C     WITH CONTRIBUTIONS TO MAIN, DATAI AND SOLVE BY S.P. LARSON        AA00050
C                                                                       AA00060*
C     REVISIONS BY D.B. SAPIK (USGS-WRD, TACOMA, WASH.) DURING 1982-83: AA00070*
C        1. ADDED SUBRS. FWSW AND ZFWSW TO LOCATE THE FRESHWATER-       AA00080*
C           SEAWATER INTERFACE (CALLED FROM MAIN ROUTINE).  ADDED NEW   AA00090*
C           STATEMENTS IN THE MAIN ROUTINE, SUBRS. CHECKI, DATAI, AND   AA00100*
C           SOLVE TO SUPPORT SUBR. FWSW.                                AA00110*
C        2. ADDED SUBR. GWDBAS TO READ DATA FOR ARRAYS USED IN THIS     AA00120*
C           MODEL (CALLED FROM SUBR. DATAI).                            AA00130*
C           DELETED ENTRY POINTS ARRAY AND MDAT IN SUBR. DATAI AND      AA00140*
C           DELETED CALLS TO ARRAY AND MDAT FROM MAIN ROUTINE.          AA00150*
C        4. ADDED STATEMENTS IN THE MAIN ROUTINE AND SUBRS. CHECKI,     AA00160*
C           DATAI, AND SOLVE THAT ARE NEEDED TO SIMULATE GROUNDWATER    AA00170*
C           FLOW BETWEEN AQUIFERS, DRAINS, AND LEAKY RIVERS.            AA00180*
C        5. ADDED SUBR. VSEC TO PRINT VALUES OF MODEL PARAMETERS        AA00190*
C           ALONG VERTICAL SECTIONS THRU THE MODEL (CALLED FROM         AA00200*
C        6. ADDED SUBR. MSQDIF TO COMPUTE ROOT-MEAN-SQUARE FOR          AA00210*
C           DIFFERENCES BETWEEN COMPUTED AND OBSERVED HEAD (CALLED FROM AA00220*
C           MAIN ROUTINE).                                              AA00230*
C        7. REVISED SUBR. CHECKI TO COMPUTE AND PRINT MASS BALANCE FOR  AA00240*
C           EACH MODEL LAYER, PRINT FLUX TO CONSTANT-HEAD, RIVER, AND   AA00250*
C           DRAIN CELLS, AND COMPUTE MASS BALANCE FOR SPECIFIED         AA00260*
C           SUBAREAS (ADDED ENTRY CHECKR TO READ COORDINATES OF         AA00270*
C           SUBAREAS).                                                  AA00280*
C        8. ADDED STATEMENTS IN SUBR. SOLVE TO CALL THE IBM LIBRARY     AA00290*
C           ROUTINE ERRSET WHEN FLOATING-POINT OPERATIONS RESULT IN     AA00300*
C           OVERFLOW, UNDERFLOW, OR DIVIDE CHECK.  SUBR. ERRSET CALLS   AA00310*
C           SUBR. ERMES1 (WRITTEN BY SAPIK) TO PRINT COORDINATES OF     AA00320*
C           MODEL CELLS WHEN ERRORS OCCUR.                              AA00330*
C     ------------------------------------------------------------------AA00340
C                                                                       AA00350
C     SPECIFICATIONS:                                                   AA00360
      REAL*8  YSTR, TFLX, CFLX, CTFLX, SUMFLX                           AA00370*
      REAL*8  XLABEL,YLABEL,TITLE,XN1,MESUR                             AA00380*
C                                                                       AA00390
      DIMENSION Y(300000), L(50), HEADNG(33), SETNAM(20)                AA00400*
C                                                                       AA00410
      EQUIVALENCE (YSTR,Y(1))                                           AA00420
C                                                                       AA00430
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,       NPER,KTH,ITMAX,LENGTH,KP,NAA00440*
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCAA00450
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN   AA00460*
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC   AA00470*
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD       AA00480*
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY                               AA00490*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1      AA00500*
     1, ERMULT, STEADY, DELT1                                           AA00510*
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)       AA00520*
```

```
      COMMON /CK/ SUMFLX(4,41), TFLX(9,2), CFLX(9,2,40), CTFLX(9,2),     AA00530*
     1 FLX(9,2,40)                                                        AA00540*
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANKAA00550*
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),AA00560*
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                           AA00570*
C                                                                         AA00580*
C...SPECIFY SIZE OF THE Y-VECTOR DIMENSIONED IN THIS ROUTINE.             AA00590*
      DATA IYSIZE / 300000 /                                              AA00600*
C                                                                         AA00610*
C...SPECIFY SIZE OF ARRAYS IN COMMON BLOCK SARRAY (LEVEL1, LEVEL2,        AA00620*
C   NODCH) AND COMMON BLOCK CK (SUMFLX, CFLX, FLX).                       AA00630*
      DATA MXLAY, MXITR / 40, 51 /                                        AA00640*
C                                                                         AA00650*
C...TEXT PASSED TO SUBRS. VSEC, MSQDIF.                                   AA00660*
      DATA  SETNAM /4HCALC,4HHEAD, 4HINTR,4HUCOD, 16*1H /                 AA00670*
     2,    IBLANK/ 1H  /                                                  AA00680*
C                                                                         AA00690*
C     ............................................................AA00700
C                                                                         AA00710*
      IERR=0                                                              AA00720*
C                                                                         AA00730*
C...READ RECORDS 1 AND 2 IN GROUP I DATA...                               AA00740*
      READ (5,200) HEADNG                                                 AA00750
      WRITE (6,190) HEADNG                                                AA00760
C                                                                         AA00770
C...READ RECORD 3 IN GROUP I DATA...                                      AA00780*
      READ (5,160)  I0, J0, K0, ITMAX, NCH, NRIV, NDRAIN, LWTABL,         AA00790*
     1 LRECH, ISALT, MSQCAL                                               AA00800*
        IF(K0.LE.0) K0=1                                                  AA00810*
        IF(ITMAX.LE.0) ITMAX=1                                            AA00820*
      WRITE (6,180) I0, J0, K0, ITMAX, NCH, NRIV, NDRAIN, LWTABL,         AA00830*
     1 LRECH, ISALT, MSQCAL                                               AA00840*
        IF(I0.LT.3 .OR. J0.LT.3) GO TO 901                                AA00850*
        IF(K0.GT.MXLAY) GO TO 903                                         AA00860*
        MXITR=MXITR-1                                                     AA00870*
        IF(ITMAX.GT.MXITR) GO TO 904                                      AA00880*
C                                                                         AA00890*
C...READ RECORD 4 IN GROUP I DATA...                                      AA00900*
      READ (5,210) IDRAW,IHEAD,IFLO,IDK1,IDK2,IWATER,IQRE,IPU1,IPU2,ITK AA00910
     1,IEQN                                                               AA00920*
      WRITE (6,220) IDRAW,IHEAD,IFLO,IDK1,IDK2,IWATER,IQRE,IPU1,IPU2,ITKAA00930
     1,IEQN                                                               AA00940*
C                                                                         AA00950*
        K=0                                                               AA00960*
        IF(IWATER.NE.ICHK(6)) GO TO 2                                     AA00970*
          K=LWTABL                                                        AA00980*
          IF(LWTABL.LE.0 .OR. LWTABL.GT.K0) K=K0                          AA00990*
    2   LWTABL=K                                                          AA01000*
C                                                                         AA01010*
        K=0                                                               AA01020*
        IF(IQRE.NE.ICHK(7)) GO TO 3                                       AA01030*
          K=LRECH                                                         AA01040*
          IF(LRECH.GT.K0) K=K0                                            AA01050*
          IF(LWTABL.GT.0 .AND. LRECH.GT.0 .AND. LRECH.GT.LWTABL)K=LWTABLAA01060*
    3   LRECH=K                                                           AA01070*
```

```
C                                                                       AA01080*
C...ALLOCATE SPACE FOR ARRAYS...                                        AA01090*
      J1=J0-1                                                           AA01100
      I1=I0-1                                                           AA01110
      K1=K0-1                                                           AA01120
      I2=I0-2                                                           AA01130
      J2=J0-2                                                           AA01140
      K2=K0-2                                                           AA01150
      NIJ=I0*J0                                                         AA01160*
      NNOD=NIJ*K0                                                       AA01170*
C-    IMAX=MAX0(I0,J0)                                                  AA01180*
C-    NCD=MAX0(1,NCH)                                                   AA01190*
      IMAX=1                                                            AA01200*
      NCD=1                                                             AA01210*
      ITMX1=ITMAX+1                                                     AA01220
      ISIZ=NNOD                                                         AA01230*
      IK1=NIJ                                                           AA01240*
      IK2=MAX0(IK1*K1,1)                                                AA01250
C...ALLOCATE SPACE FOR THE ARRAYS PHI, STRT, OLD, T, S, TR, TC, TK,     AA01260*
C   WELL, EL, FL, GL, V, XI.                                            AA01270*
      ISUM=2*ISIZ+1                                                     AA01280
      L(1)=1                                                            AA01290
      DO 30 I=2,14                                                      AA01300
      IF (I.NE.8) GO TO 20                                              AA01310
      L(8)=ISUM                                                         AA01320
      ISUM=ISUM+IK2                                                     AA01330
      IF (IK2.EQ.1) GO TO 10                                            AA01340
      IK=I0                                                             AA01350
      JK=J0                                                             AA01360
      K5=K1                                                             AA01370
      GO TO 30                                                          AA01380
   10 IK=1                                                              AA01390
      JK=1                                                              AA01400
      K5=1                                                              AA01410
      GO TO 30                                                          AA01420
   20 L(I)=ISUM                                                         AA01430
      ISUM=ISUM+ISIZ                                                    AA01440
   30 CONTINUE                                                          AA01450
C...ALLOCATE SPACE FOR THE ARRAY DELX.                                  AA01460*
      L(15)=ISUM                                                        AA01470
      ISUM=ISUM+J0                                                      AA01480
C...ALLOCATE SPACE FOR THE ARRAY DELY.                                  AA01490*
      L(16)=ISUM                                                        AA01500
      ISUM=ISUM+I0                                                      AA01510
C...ALLOCATE SPACE FOR THE ARRAY DELZ.                                  AA01520*
      L(17)=ISUM                                                        AA01530
      ISUM=ISUM+K0                                                      AA01540
C...ALLOCATE SPACE FOR THE ARRAY DDN.                                   AA01550*
      L(18)=ISUM                                                        AA01560
      ISUM=ISUM+IMAX                                                    AA01570
C...ALLOCATE SPACE FOR THE ARRAY FACT.                                  AA01580*
      L(19)=ISUM                                                        AA01590
      ISUM=ISUM+K0*3                                                    AA01600
C...ALLOCATE SPACE FOR THE ARRAY TEST3.                                 AA01610*
      L(20)=ISUM                                                        AA01620
```

```
      ISUM=ISUM+ITMX1                                                  AA01630
C...ALLOCATE SPACE FOR THE ARRAY JFLO.                                 AA01640*
      L(21)=ISUM                                                       AA01650
      ISUM=ISUM+3*NCD                                                  AA01660
C...ALLOCATE SPACE FOR THE ARRAY FLOW.                                 AA01670*
      L(22)=ISUM                                                       AA01680
      ISUM=ISUM+NCD                                                    AA01690
C...ALLOCATE SPACE FOR THE ARRAYS PERM AND BOTTOM.                     AA01700*
      L(23)=ISUM                                                       AA01710
      IF (IWATER.NE.ICHK(6)) GO TO 40                                  AA01720
      ISUM=ISUM+IK1                                                    AA01730
      L(24)=ISUM                                                       AA01740
      ISUM=ISUM+IK1                                                    AA01750
      IP=I0                                                            AA01760
      JP=J0                                                            AA01770
      GO TO 50                                                         AA01780
   40 ISUM=ISUM+1                                                      AA01790
      L(24)=ISUM                                                       AA01800
      ISUM=ISUM+1                                                      AA01810
      IP=1                                                             AA01820
      JP=1                                                             AA01830
C...ALLOCATE SPACE FOR THE ARRAY QRE.                                  AA01840*
   50 L(25)=ISUM                                                       AA01850
      IF (IQRE.NE.ICHK(7)) GO TO 60                                    AA01860
      ISUM=ISUM+IK1                                                    AA01870
      IQ=I0                                                            AA01880
      JQ=J0                                                            AA01890
      GO TO 70                                                         AA01900
   60 ISUM=ISUM+1                                                      AA01910
      IQ=1                                                             AA01920
      JQ=1                                                             AA01930
C...ALLOCATE SPACE FOR THE ARRAYS IDR, RH, RC, RB.                     AA01940*
   70 L(26)=ISUM                                                       AA01950*
      IF(NRIV.EQ.0) GO TO 75                                           AA01960*
      ISUM=ISUM+ISIZ                                                   AA01970*
      L(27)=ISUM                                                       AA01980*
      ISUM=ISUM+NRIV                                                   AA01990*
      L(28)=ISUM                                                       AA02000*
      ISUM=ISUM+NRIV                                                   AA02010*
      L(29)=ISUM                                                       AA02020*
      ISUM=ISUM+NRIV                                                   AA02030*
      IA1=I0                                                           AA02040*
      JA1=J0                                                           AA02050*
      KA1=K0                                                           AA02060*
      GO TO 79                                                         AA02070*
   75 L(27)=ISUM+1                                                     AA02080*
      L(28)=ISUM+2                                                     AA02090*
      L(29)=ISUM+3                                                     AA02100*
      ISUM=ISUM+4                                                      AA02110*
      IA1=1                                                            AA02120*
      JA1=1                                                            AA02130*
      KA1=1                                                            AA02140*
C...ALLOCATE SPACE FOR THE ARRAYS IDN, DRCOF, DRELV.                   AA02150*
   79 L(30)=ISUM                                                       AA02160*
      IF (NDRAIN.LE.0) GO TO 80                                        AA02170*
```

```
      ISUM=ISUM+ISIZ                                                    AA02180*
      L(31)=ISUM                                                        AA02190*
      ISUM=ISUM+NDRAIN                                                  AA02200*
      L(32)=ISUM                                                        AA02210*
      ISUM=ISUM+NDRAIN                                                  AA02220*
      IA2=I0                                                            AA02230*
      JA2=J0                                                            AA02240*
      KA2=K0                                                            AA02250*
      GO TO 85                                                          AA02260*
   80 L(31)=ISUM+1                                                      AA02270*
      L(32)=ISUM+2                                                      AA02280*
      ISUM=ISUM+3                                                       AA02290*
      IA2=1                                                             AA02300*
      JA2=1                                                             AA02310*
      KA2=1                                                             AA02320*
C...ALLOCATE SPACE FOR THE ARRAY LAYRCH.                                 AA02330*
   85 L(33)=ISUM                                                        AA02340*
      J=1                                                               AA02350*
      IF(IQRE.EQ.ICHK(7)) J=NIJ                                         AA02360*
      ISUM=ISUM+J                                                       AA02370*
C...ALLOCATE SPACE FOR THE ARRAYS INT, TT, DBOT, BTM, ISEA, RC2, WEL2,   AA02380*
C   ZOLD.                                                               AA02390*
   90 L(34)=ISUM                                                        AA02400*
      IF(ISALT.LE.0) GO TO 92                                           AA02410*
      ISUM=ISUM+ISIZ                                                    AA02420*
      L(35)=ISUM                                                        AA02430*
      ISUM=ISUM+ISIZ                                                    AA02440*
      L(36)=ISUM                                                        AA02450*
      ISUM=ISUM+ISIZ                                                    AA02460*
      L(37)=ISUM                                                        AA02470*
      J=1                                                               AA02480*
      IF(IWATER.EQ.ICHK(6)) J=NIJ                                       AA02490*
      ISUM=ISUM+J                                                       AA02500*
      L(38)=ISUM                                                        AA02510*
      ISUM=ISUM+ISIZ                                                    AA02520*
      L(39)=ISUM                                                        AA02530*
      J=MAX0(1, NRIV)                                                   AA02540*
      ISUM=ISUM+J                                                       AA02550*
      L(40)=ISUM                                                        AA02560*
      ISUM=ISUM+ISIZ                                                    AA02570*
      L(41)=ISUM                                                        AA02580*
      ISUM=ISUM+ISIZ                                                    AA02590*
      IA3=I0                                                            AA02600*
      JA3=J0                                                            AA02610*
      KA3=K0                                                            AA02620*
      GO TO 95                                                          AA02630*
   92 L(35)=ISUM+1                                                      AA02640*
      L(36)=ISUM+2                                                      AA02650*
      L(37)=ISUM+3                                                      AA02660*
      L(38)=ISUM+4                                                      AA02670*
      L(39)=ISUM+5                                                      AA02680*
      L(40)=ISUM+6                                                      AA02690*
      L(41)=ISUM+7                                                      AA02700*
      ISUM=ISUM+8                                                       AA02710*
      IA3=1                                                             AA02720*
```

```
      JA3=1                                                             AA02730*
      KA3=1                                                             AA02740*
   95 CONTINUE                                                          AA02750*
C                                                                       AA02760*
      ISUM=ISUM-1                                                       AA02770*
      WRITE (6,170) ISUM, IYSIZE                                        AA02780*
      IF (ISUM.GT.IYSIZE) GO TO 902                                     AA02790*
C                                                                       AA02800
C     ---PASS INITIAL ADDRESSES OF ARRAYS TO SUBROUTINES---             AA02810
      CALL DATAI(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))AA02820
     1,Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(23)),Y(L(AA02830
     224)),Y(L(25)),Y(L(33)),Y(L(13)),                                  AA02840*
     3Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)),Y(L(31)),Y(L(32)),   AA02850*
     3Y(L(38)) )                                                        AA02860*
      CALL STEP(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(10)) )              AA02870*
      CALL SOLVE(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))AA02880
     1,Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(10)),Y(L(AA02890
     211)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(20)),Y(L(25)),Y(L(33)),       AA02900*
     3Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)),Y(L(31)),Y(L(32)),   AA02910*
     4Y(L(38)) )                                                        AA02920*
      CALL COEF(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)),AA02930
     1Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(23)),Y(L(2AA02940
     24)) )                                                             AA02950*
      CALL CHECKI(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)AA02960
     1),Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(21)),Y(LAA02970
     2(22)),Y(L(25)),Y(L(33)),                                          AA02980*
     3Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)),Y(L(31)),Y(L(32)),   AA02990*
     4Y(L(38)),Y(L(10)),Y(L(11)),Y(L(12)) )                             AA03000*
      CALL PRNTAI(Y(L(1)),Y(L(2)),Y(L(4)),Y(L(5)),Y(L(9)),Y(L(15)),Y(L(1AA03010
     16)))                                                              AA03020
C                                                                       AA03030
C...READ RECORD 5 IN GROUP I DATA...                                    AA03040*
      READ (5,160) NVSEC, NSUBAL, IBALAY, IPFLX, ISVFLX, ISVHED, ISVHDX AA03050*
        IF(IPFLX.GT.1) IPFLX=1                                          AA03060*
      WRITE(6,202) NVSEC, NSUBAL, IBALAY, IPFLX, ISVFLX, ISVHED, ISVHDX AA03070*
C                                                                       AA03080*
C...READ RECORDS 6 AND 7 IN GROUP I DATA: DEFINE VERTICAL SECTIONS      AA03090*
C        FOR PRINTING HEAD AND INTRUSION CODES...                       AA03100*
      IF(NVSEC.GT.0) CALL VSEC (0, SETNAM(1), Y(1), Y(2), Y(3) )        AA03110*
C                                                                       AA03120*
C...READ RECORD 8 IN GROUP I DATA: COORDINATES OF SUBAREAS FOR MASS     AA03130*
C         BALANCE COMPUTATIONS...                                       AA03140*
      CALL CHECKR                                                       AA03150*
C                                                                       AA03160*
C...READ MORE GROUP I DATA AND READ DATA FOR GROUPS II AND V...         AA03170*
      CALL DATAIN                                                       AA03180
C                                                                       AA03190*
C...INITIALIZE THE ROUTINE THAT PRINTS SCALED ALPHANUMERIC MAPS.        AA03200*
      IF (XSCALE.GT.0.) CALL MAP                                        AA03210*
C                                                                       AA03220*
C     ---COMPUTE T COEFFICIENTS---                                      AA03230
      NTR=2                                                             AA03240*
      IF (IWATER.NE.ICHK(6)) GO TO 120                                  AA03250*
        CALL TRANS                                                      AA03260*
        NTR=1                                                           AA03270*
```

```
  120 CALL TCOF
      NTR=0
      IF (ISALT.GT.0) NTR=2
      IF (ISALT.GT.0 .AND. IWATER.EQ.ICHK(6)) NTR=1
C
C...READ GROUP III DATA FOR FRESHWATER-SALTWATER PROBLEM.
      NTAD=1
      NTRIAL=0
     IF(ISALT.GT.0)
    1 CALL FWSW (0, Y(L(1)), Y(L(4)), Y(L(9)), Y(L(23)),
    2    Y(L(24)), Y(L(26)), Y(L(27)), Y(L(28)), Y(L(34)),
    3    Y(L(35)), Y(L(36)), Y(L(37)), Y(L(38)), Y(L(39)),
    4    Y(L(40)), Y(L(41)), Y(L(5)), Y(L(10)), Y(L(11)), Y(L(12)),
    5    Y(L(30)) )
C
C...READ GROUP VI DATA: PARAMETERS FOR COMPUTING MEAN-SQUARE ERROR
C   BETWEEN OBSERVED AND MODEL-COMPUTED HEAD.
      IF(MSQCAL.GT.0) CALL MSQDIF (Y(1), Y(2), Y(3), Y(4), SETNAM(1),
    1   SUM)
C
C     ---COMPUTE ITERATION PARAMETERS---
      CALL ITER
      ERRSAV=ERR
C
C...READ GROUP IV DATA: PARAMETERS FOR START OF A NEW STRESS PERIOD...
  140 CALL NEWPER
C
      KT=0
      IFINAL=0
      ISTEDY=0
C
C     ---START NEW TIME STEP COMPUTATIONS---
  150 CALL NEWSTP
      IERR=0
C
C     ---START NEW ITERATION IF MAXIMUM NO. ITERATIONS NOT EXCEEDED---
      CALL NEWITA
      IF(KT.GE.NUMT) IFINAL=1
      IF(KT.GE.NUMT .AND. KP.GE.NPER .OR. IERR.EQ.2) IFINAL=2
C
C     ---COMPUTE MASS BALANCE---
      CALL CHECK
C
C     ---PRINT RESULTS AT END OF TIME STEP---
      IF(ISALT.LE.0 .OR. IERR.GT.0) CALL OUTPUT
C
C     ---LAST TIME STEP IN STRESS PERIOD ?---
      IF (IFINAL.EQ.0) GO TO 150
C
C...LOCATE POSITION OF FRESHWATER-SALTWATER INTERFACE.
      IF(ISALT.LE.0 .OR. IERR.GT.0) GO TO 158
      NTRIAL=NTRIAL+1
        CALL FWSW (1, Y(L(1)), Y(L(4)), Y(L(9)), Y(L(23)),
    2    Y(L(24)), Y(L(26)), Y(L(27)), Y(L(28)), Y(L(34)),
    3    Y(L(35)), Y(L(36)), Y(L(37)), Y(L(38)), Y(L(39)),
```

```
     4     Y(L(40)), Y(L(41)), Y(L(5)), Y(L(10)), Y(L(11)), Y(L(12)),
     5     Y(L(30)) )
      IF(IERR.EQ.0 .OR. NTRIAL.GE.NTAD) GO TO 157
        IF(WMAX1.LE.0.) CALL ITER
        ERR=ERR*ERMULT
        KT=0
        IFINAL=0
        DELT=DELT1
        SUM=0.
        SUMP=0.
        GO TO 150
C
  157 IF(IERR.GT.0) WRITE(6,9001)
        CALL OUTPUT
        NTRIAL=0
  158 ERR=ERRSAV
C
C     ---COMPUTE ROOT-MEAN-SQUARE FOR DIFFERENCES BETWEEN OBSERVED
C        AND COMPUTED HEAD---
      IF(MSQCAL.EQ.1) CALL MSQDIF (Y(1), Y(L(4)), Y(L(15)), Y(L(16)),
     1   SETNAM(1), SUM)
C
C     ---CHECK FOR NEW STRESS PERIOD---
      IF (KP.LT.NPER .AND. IERR.EQ.0) GO TO 140
C
C     ---COMPUTE ROOT-MEAN-SQUARE FOR DIFFERENCES BETWEEN OBSERVED
C        AND COMPUTED HEAD---
      IF(MSQCAL.EQ.2) CALL MSQDIF (Y(1), Y(L(4)), Y(L(15)), Y(L(16)),
     1   SETNAM(1), SUM)
C
C...PRINT DATA ON FRESHWATER-SALTWATER INTERFACE.
      IF(ISALT.GT.0)
     1 CALL FWSW (2, Y(L(1)), Y(L(4)), Y(L(9)), Y(L(23)),
     2     Y(L(24)), Y(L(26)), Y(L(27)), Y(L(28)), Y(L(34)),
     3     Y(L(35)), Y(L(36)), Y(L(37)), Y(L(38)), Y(L(39)),
     4     Y(L(40)), Y(L(41)), Y(L(5)), Y(L(10)), Y(L(11)), Y(L(12)),
     5     Y(L(30)) )
C
C...PRINT VALUES OF MODEL PARAMETERS ALONG VERTICAL SECTIONS.
      IF(NVSEC.LE.0) GO TO 165
      CALL VSEC(1, SETNAM(1), Y(L(10)), Y(L(13)), Y(L(4)) )
      IF(ISALT.GT.0) CALL VSEC (2, SETNAM(3), Y(L(11)), Y(L(13)),
     1   Y(L(4)) )
  165 CONTINUE
C
      GO TO 999

C
C
C...ERROR MESSAGES...
  901 WRITE(6,9901)
        GO TO 999
  902 WRITE(6,9902) IYSIZE
        GO TO 999
  903 WRITE(6,9903) MXLAY
```

```
        GO TO 999
  904 WRITE(6,9904) MXITR
        GO TO 999
C
C...END OF JOB...
  999 WRITE(6,9999)
C
      STOP
C
  160 FORMAT(16I5)
  170 FORMAT(  '-*** WORDS USED FOR Y-VECTOR IN MAIN ROUTINE = ',I8/
     1      1H0,4X, 'DIMENSIONED SIZE FOR Y-VECTOR',10X, 3H = ,I8)
  180 FORMAT(1H-,  'DATA FROM RECORD 3 IN GROUP I DATA:'/1X,35(1H-)/
     1  6X, 13HROWS IN MODEL,37X, 3H = ,I5/
     2  6X, 16HCOLUMNS IN MODEL,34X, 3H = ,I5 /
     3  6X, 15HLAYERS IN MODEL,35X, 3H = ,I5 /
     4  6X, 46HMAX. ITERATIONS FOR EQUATION-SOLVING PROCEDURE,4X,
     5         3H = ,I5 /
     6  6X, 31HMAX. NO. OF CONSTANT-HEAD CELLS,19X, 3H = ,I5 /
     7  6X, 17HLEAKY RIVER CELLS,33X, 3H = ,I5 /
     8  6X, 11HDRAIN CELLS,39X, 3H = ,I5 /
     9  6X, 29HLAYER FOR WATER-TABLE AQUIFER,21X, 3H = ,I5/
     X  6X, 28HMETHOD OF INCLUDING RECHARGE,22X, 3H = ,I5/
     1  6X, 29HSOLVE FOR SALTWATER INTERFACE,21X, 3H = ,I5 /
     2  6X, 45HCOMPUTE ROOT-MEAN-SQUARE FOR HEAD DIFFERENCES,5X,3H = ,
     3 I5 )
  190 FORMAT ('1',33A4)
  200 FORMAT (20A4)
  202 FORMAT( '-DATA FROM RECORD 5 IN GROUP I DATA:'/ 1X,35(1H-)/
     1 6X, 'NVSEC  = ',I5 / 6X, 'NSUBAL = ',I5 / 6X, 'IBALAY = ',I5 /
     2 6X, 'IPFLX  = ',I5 / 6X, 'ISVFLX = ',I5 / 6X, 'ISVHED = ',I5 /
     3 6X, 'ISVHDX = ',I5 )
  210 FORMAT (16(A4,1X))
  220 FORMAT ('-DATA FROM RECORD 4 IN GROUP I DATA:'/1X,35(1H-)/
     1 6X, 'IDRAW  = ',A4 / 6X, 'IHEAD  = ',A4 / 6X, 'IFLO   = ',A4 /
     2 6X, 'IDK1   = ',A4 / 6X, 'IDK2   = ',A4 / 6X, 'IWATER = ',A4 /
     3 6X, 'IQRE   = ',A4 / 6X, 'IPU1   = ',A4 / 6X, 'IPU2   = ',A4 /
     4 6X, 'ITK    = ',A4 / 6X, 'IEQN   = ',A4 )
 9001 FORMAT(  '-***ERROR*** SOLUTION NOT OBTAINED FOR FRESHWATER-',
     1 'SALTWATER INTERFACE.')
 9901 FORMAT('-***ERROR*** VALUE OF I0 OR J0 FOR MODEL SIZE IS LESS'
     1,  ' THAN 3.')
 9902 FORMAT('-***ERROR*** WORDS USED FOR Y-VECTOR EXCEED DIMENSIONED',
     1  ' SIZE OF ',I8, 1H. )
 9903 FORMAT(  '-***ERROR*** VALUE OF K0 (MODEL LAYERS) EXCEEDS ',I3,
     1  ' (MAX. SIZE FOR ARRAYS IN COMMON BLOCKS SARRAY AND CK).' )
 9904 FORMAT(  '-***ERROR*** VALUE OF ITMAX (MAX. ITERATIONS) EXCEEDS ',
     1 I3,  ' (MAX. SIZE FOR THE ARRAY NODCH).' )
 9999 FORMAT('-*** END OF JOB ***' //)
      END
```

```
      BLOCK DATA                                                        BD00010
C     ----------                                                        BD00020
C                                                                       BD00030
C     SPECIFICATIONS:                                                   BD00040
      REAL*8  XLABEL,YLABEL,TITLE,XN1,MESUR                             BD00050
C                                                                       BD00060
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)       BD00070*
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANKBD00080
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),BD00090
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                          BD00100
C     ******************************************************************BD00110
C                                                                       BD00120
      DATA ICHK/'DRAW','HEAD','MASS','DK1','DK2','WATE','RECH','PUN1','PBD00130
     1UN2','ITKR','EQN3',9*' '/                                         BD00140*
      DATA SYM/'1','2','3','4','5','6','7','8','9','0','*','|','-','+','BD00150
     1 ','R','W'/                                                       BD00160
      DATA PRNT/122*' '/,N1,N2,N3,XN1/6,10,133,.833333333D-1/,BLANK/60*'BD00170
     1 '/,NA(4)/1000/                                                   BD00180
      DATA XLABEL/' X DIS- ','TANCE IN',' MILES '/,YLABEL/'DISTANCE',' BD00190
     1FROM OR','IGIN IN ','Y DIRECT','ION, IN ','MILES   '/,TITLE/'PLOT BD00200
     2OF ','DRAWDOWN',' ','PLOT OF ','HYDRAULI','C HEAD'/               BD00210
      DATA DIGIT/'1','2','3','4','5','6','7','8','9','10','11','12','13'BD00220
     1,'14','15','16','17','18','19','20','21','22','23','24','25','26',BD00230
     2'27','28','29','30','31','32','33','34','35','36','37','38','39','BD00240
     340','41','42','43','44','45','46','47','48','49','50','51','52','5BD00250
     43','54','55','56','57','58','59','60','61','62','63','64','65','66BD00260
     5','67','68','69','70','71','72','73','74','75','76','77','78','79'BD00270*
     6,'80','81','82','83','84','85','86','87','88','89','90','91','92' BD00280
     7,'93','94','95','96','97','98','99','100','101','102','103','104' BD00290*
     8,'105','106','107','108','109','110','111','112','113','114','115'BD00300
     9,'116','117','118','119','120','121','122'/                       BD00310
      DATA VF1/'(1H ',',',','    ','A1,F','10.2',')'/                   BD00320
      DATA VF2/'(1H ',',',','    ','A1,1','X,A8',')'/                   BD00330
      DATA VF3/'(1H0',',',','    ','A1,F','3.1,','12F1','0.2)'/         BD00340
C     ******************************************************************BD00350
      END                                                               BD00360
```

```
      SUBROUTINE CHECKI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FA
     1CT,JFLO,FLOW,QRE,LAYRCH, IDR,RH,RC,RB, IDN,DRCOF,DRELV, ISEA,
     2EL,FL,GL)
C     ------------------------------------------------------------------
C     COMPUTE A VOLUMETRIC BALANCE
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL*8  PHI, TFLX, CFLX, CTFLX, SUMFLX, NAME(20)
C
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0
     .1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(1,3,K0), JFLO(NCD,3),
     3 FLOW(NCD), QRE(IQ,JQ), IDR(IA1,JA1,KA1), RH(1), RC(1), RB(1),
     4 IDN(IA2,JA2,KA2), DRCOF(1), DRELV(1), LAYRCH(IQ,JQ),ISEA(IA3,JA3,
     5KA3),EL(I0,J0,K0),FL(I0,J0,K0),GL(I0,J0,K0),SETID(6)
     6,IBT(10),IBB(10),IBL(10),IBR(10), AFLX(33)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,       NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1
     1, ERMULT, STEADY, DELT1
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)
C
C...THE FOLLOWING PARAMETERS ARE STORAGE AREAS FOR ELEMENTS IN THE MASS
C   BALANCE TABLE:
C     FLX    = FLUX RATES FOR EACH LAYER IN THIS TIME STEP.
C     TFLX   = FLUX RATES SUMMED FOR ALL LAYERS IN THIS TIME STEP.
C     CFLX   = CUMULATIVE FLUX VOLUMES SUMMED FOR EACH LAYER SINCE THE
C              START OF SIMULATION.
C     CTFLX  = CUMULATIVE FLUX VOLUMES SUMMED FOR ALL LAYERS SINCE THE
C              START OF SIMULATION.
C     SUMFLX = TOTALS FOR FLUXES IN EACH COLUMN OF MASS BALANCE TABLES.
C       NOTE THAT TFLX, CFLX, AND CTFLX ARE DOUBLE-PRECISION PARAMETERS.
      COMMON /CK/ SUMFLX(4,41), TFLX(9,2), CFLX(9,2,40), CTFLX(9,2),
     1 FLX(9,2,40)
C
C...THE FOLLOWING PARAMETERS DEFINE THE SIZE OF THE MASS BALANCE TABLES
C   AND THE DIMENSIONS OF ARRAYS IN COMMON BLOCK CK:
C     NFLX   = NO. OF ROWS IN MASS BALANCE TABLE (SEE ARRAYS FLX, TFLX,
C              CFLX, AND CTFLX).
      DATA  NFLX / 9 /
C
C...THE FOLLOWING PARAMETERS DEFINE ROW NOS. IN THE MASS BALANCE TABLES
C   FOR THE FOLLOWING MASS BALANCE COMPONENTS: STORAGE (JSTR),
C   RECHARGE (JRCH), CONSTANT FLUX (JCFX), PUMPING (JPMP), CONSTANT
C   HEAD (JCHD), LEAKY RIVERS (JRIV), DRAINS (JDRN), VERTICAL LEAKAGE
C   ACROSS TOP OF LAYER (JLKT), VERTICAL LEAKAGE ACROSS BOTTOM OF
C   LAYER (JLKB):
      DATA  JSTR, JRCH, JCFX, JPMP, JCHD, JRIV, JDRN, JLKT, JLKB
     1  / 1, 2, 3, 4, 5, 6, 7, 8, 9 /
```

```
C                                                                       CK00560*
C...NAMES FOR COMPONENTS IN THE MASS BALANCE TABLES:                    CK00570*
      DATA  NAME / 7HSTORAGE,1H , 8HRECHARGE,1H , 8HCONST. F,3HLUX,     CK00580*
     1  7HPUMPING,1H , 8HCONST. H,3HEAD, 6HRIVERS,1H , 6HDRAINS,1H ,    CK00590*
     2  8HACROSS T,2HOP, 8HACROSS B,3HOT., 8HSUBTOTAL,1HS /             CK00600*
C                                                                       CK00610*
C...SET VALUE FOR MAXIMUM NO. OF SUBAREAS USED IN MASS BAL. COMPUTATIONSCK00620*
      DATA  MXSUB / 10 /                                                CK00630*
C                                                                       CK00640*
C...NAMES FOR MASS BALANCE DATA SETS SAVED ON DISK FILE.                CK00650*
      DATA SETID /4HCONS,4HTHED, 4HRIVE,1HR, 4HDRAI,1HN/                CK00660*
C                                                                       CK00670*
      GO TO 998                                                         CK00680*
C-----------------------------------------------------------------------CK00690*
C     ********************                                              CK00700*
      ENTRY CHECKR                                                      CK00710*
C     ********************                                              CK00720*
C...READ RECORD 8 IN GROUP I DATA: COORDINATES OF SUBAREAS FOR MASS     CK00730*
C        BALANCE COMPUTATIONS...                                        CK00740*
        IBT(1)=2                                                        CK00750*
        IBB(1)=I1                                                       CK00760*
        IBL(1)=2                                                        CK00770*
        IBR(1)=J1                                                       CK00780*
        NSUB=MAX0(1, NSUBAL)                                            CK00790*
      IF(NSUBAL.LE.0) GO TO 22                                          CK00800*
        IF(NSUBAL.GT.MXSUB) GO TO 901                                   CK00810*
        READ(5,8000) (IBT(J),IBB(J),IBL(J),IBR(J),J=1,NSUBAL)           CK00820*
        WRITE(6,9010)                                                   CK00830*
        IERR=0                                                          CK00840*
        DO 21 J=1,NSUBAL                                                CK00850*
          WRITE(6,9011) J, IBT(J),IBB(J),IBL(J),IBR(J)                  CK00860*
          IF(.NOT.                                                      CK00870*
     1      (IBT(J).LE.0 .OR. IBT(J).GT.IBB(J) .OR. IBB(J).GT.I0 .OR.   CK00880*
     1       IBL(J).LE.0 .OR. IBL(J).GT.IBR(J) .OR. IBR(J).GT.J0) )     CK00890*
     2       GO TO 21                                                   CK00900*
           WRITE(6,9020)                                                CK00910*
           IERR=1                                                       CK00920*
   21   CONTINUE                                                        CK00930*
        IF(IERR.GT.0) GO TO 997                                         CK00940*
   22 CONTINUE                                                          CK00950*
      GO TO 998                                                         CK00960*
C                                                                       CK00970*
C-----------------------------------------------------------------------CK00980*
C                                                                       CK00990*
C     ********************                                              CK01000
      ENTRY CHECK                                                       CK01010
C     ********************                                              CK01020
C     ---INITIALIZE VARIABLES---                                        CK01030
        NFLX2=JLKT-1                                                    CK01040*
        KK1=K0+1                                                        CK01050*
      DO 15 K=1,KK1                                                     CK01060*
        IF(K.EQ.KK1) GO TO 14                                           CK01070*
      DO 13 J=1,2                                                       CK01080*
      DO 13 I=1,NFLX                                                    CK01090*
        FLX(I,J,K)=0.                                                   CK01100*
```

```
        IF(K.EQ.1) TFLX(I,J)=0.                                          CK01110*
        IF(ISALT.LE.0.) GO TO 13                                         CK01120*
          CFLX(I,J,K)=0.                                                 CK01130*
          IF(K.EQ.1) CTFLX(I,J)=0.                                       CK01140*
   13 CONTINUE                                                           CK01150*
   14 DO 15 J=1,4                                                        CK01160*
        SUMFLX(J,K)=0.                                                   CK01170*
   15 CONTINUE                                                           CK01180*
C                                                                        CK01190*
      DO 17 K=1,K0                                                       CK01200*
      DO 17 J=1,J0                                                       CK01210*
      DO 17 I=1,I0                                                       CK01220*
        EL(I,J,K)=0.                                                     CK01230*
        FL(I,J,K)=0.                                                     CK01240*
        GL(I,J,K)=0.                                                     CK01250*
   17 CONTINUE                                                           CK01260*
C                                                                        CK01270*
      IPRFLX=0                                                           CK01280*
      IF(IPFLX.LE.0 .OR. ((IFLO.NE.ICHK(3) .OR. MOD(KT,KTH).NE.0) .AND. CK01290*
     1    IFINAL.NE.2)) GO TO 18                                         CK01300*
        IPRFLX=1                                                         CK01310*
        IDSK=13                                                          CK01320*
        WRITE(IDSK,9110) KP, KT, SUM                                     CK01330*
   18 CONTINUE                                                           CK01340*
C                                                                        CK01350*
C................................................                        CK01360*
C...COMPUTE FLUX RATES FOR THIS TIME STEP.                               CK01370*
      DO 220 K=1,K0                                                      CK01380
        DZ=DELZ(K)                                                       CK01390*
        KM1=K-1                                                          CK01400*
        KP1=K+1                                                          CK01410*
      DO 220 NS=1,NSUB                                                   CK01420*
        IB1=IBT(NS)                                                      CK01430*
        IB2=IBB(NS)                                                      CK01440*
        JB1=IBL(NS)                                                      CK01450*
        JB2=IBR(NS)                                                      CK01460*
      DO 220 I=IB1,IB2                                                   CK01470*
        DY=DELY(I)                                                       CK01480*
        IM1=I-1                                                          CK01490*
        IP1=I+1                                                          CK01500*
      DO 220 J=JB1,JB2                                                   CK01510*
      IF (T(I,J,K).LE.0.) GO TO 220                                      CK01520
      DX=DELX(J)                                                         CK01530*
      AREA=DX*DY                                                         CK01540*
      AX=AREA                                                            CK01550*
      BX=1.                                                              CK01560*
      IF (IEQN.NE.ICHK(11)) GO TO 2                                      CK01570*
      AX=AREA*DZ                                                         CK01580*
      BX=DZ                                                              CK01590*
    2 PHIX=PHI(I,J,K)                                                    CK01600*
      ICHD=0                                                             CK01610*
      IPRT=0                                                             CK01620*
      FLXN=0.                                                            CK01630*
      FLXP=0.                                                            CK01640*
C                                                                        CK01650*
```

```
      IF (S(I,J,K).GE.0.) GO TO 180                                     CK01660
C                                                                       CK01670
C     ---CONSTANT HEAD BOUNDARIES---                                    CK01680*
      ICHD=1                                                            CK01690*
      JM1=J-1                                                           CK01700*
      JP1=J+1                                                           CK01710*
C                                                                       CK01720*
      IF (S(I,JM1,K).LT.0..OR.T(I,JM1,K).EQ.0.) GO TO 30                CK01730*
      X=(PHIX-SNGL(PHI(I,JM1,K)))*TR(I,JM1,K)*DY*BX                     CK01740*
      IF (IPRFLX.EQ.0) GO TO 5                                          CK01750*
      WRITE (IDSK,9111) K,I,J, K,I,JM1, X                               CK01760*
      IPRT=1                                                            CK01770*
    5 IF (X) 10,30,20                                                   CK01780*
   10 FLXN=FLXN-X                                                       CK01790*
      GO TO 30                                                          CK01800*
   20 FLXP=FLXP+X                                                       CK01810*
C                                                                       CK01820*
   30 IF (S(I,JP1,K).LT.0..OR.T(I,JP1,K).EQ.0.) GO TO 51                CK01830*
      X=(PHIX-SNGL(PHI(I,JP1,K)))*TR(I,J,K)*DY*BX                       CK01840*
      IF (IPRFLX.EQ.0) GO TO 35                                         CK01850*
      IF (IPRT.GT.0) GO TO 34                                           CK01860*
      WRITE (IDSK,9111) K,I,J, K,I,JP1, X                               CK01870*
      IPRT=1                                                            CK01880*
      GO TO 35                                                          CK01890*
   34 WRITE (IDSK,9112) K,I,JP1, X                                      CK01900*
   35 IF (X) 40,51,50                                                   CK01910*
   40 FLXN=FLXN-X                                                       CK01920*
      GO TO 51                                                          CK01930*
   50 FLXP=FLXP+X                                                       CK01940*
   51 CONTINUE                                                          CK01950*
C                                                                       CK01960*
  120 IF (S(IM1,J,K).LT.0..OR.T(IM1,J,K).EQ.0.) GO TO 150               CK01970*
      X=(PHIX-SNGL(PHI(IM1,J,K)))*TC(IM1,J,K)*DX*BX                     CK01980*
      IF (IPRFLX.EQ.0) GO TO 125                                        CK01990*
      IF (IPRT.GT.0) GO TO 124                                          CK02000*
      WRITE (IDSK,9111) K,I,J, K,IM1,J, X                               CK02010*
      IPRT=1                                                            CK02020*
      GO TO 125                                                         CK02030*
  124 WRITE (IDSK,9112) K,IM1,J, X                                      CK02040*
  125 IF (X) 130,150,140                                                CK02050*
  130 FLXN=FLXN-X                                                       CK02060*
      GO TO 150                                                         CK02070*
  140 FLXP=FLXP+X                                                       CK02080*
C                                                                       CK02090*
  150 IF (S(IP1,J,K).LT.0..OR.T(IP1,J,K).EQ.0.) GO TO 171               CK02100*
      X=(PHIX-SNGL(PHI(IP1,J,K)))*TC(I,J,K)*DX*BX                       CK02110*
      IF (IPRFLX.EQ.0) GO TO 155                                        CK02120*
      IF (IPRT.GT.0) GO TO 154                                          CK02130*
      WRITE (IDSK,9111) K,I,J, K,IP1,J, X                               CK02140*
      IPRT=1                                                            CK02150*
      GO TO 155                                                         CK02160*
  154 WRITE (IDSK,9112) K,IP1,J, X                                      CK02170*
  155 IF (X) 160,171,170                                                CK02180*
  160 FLXN=FLXN-X                                                       CK02190*
      GO TO 171                                                         CK02200*
```

```
  170 FLXP=FLXP+X                                                       CK02210*
  171 CONTINUE                                                          CK02220*
C                                                                       CK02230*
      GO TO 218                                                         CK02240*
C                                                                       CK02250*
C     ---RECHARGE---                                                    CK02260*
  180 IF(IQRE.EQ.ICHK(7).AND.K.EQ.LAYRCH(I,J)) FLX(JRCH,1,K)=           CK02270*
     1   FLX(JRCH,1,K)+QRE(I,J)*AX                                      CK02280*
C                                                                       CK02290*
C     ---PUMPAGE AND CONSTANT FLUX---                                   CK02300*
      X=WELL(I,J,K)*AX                                                  CK02310*
      IF(X) 200, 202, 201                                               CK02320*
  200 FLX(JPMP,2,K)=FLX(JPMP,2,K)-X                                     CK02330*
        GO TO 202                                                       CK02340*
  201 FLX(JCFX,1,K)=FLX(JCFX,1,K)+X                                     CK02350*
  202 CONTINUE                                                          CK02360*
C                                                                       CK02370*
C     ---CHANGE IN STORAGE---                                           CK02380*
      X=S(I,J,K)*(OLD(I,J,K)-PHIX)*AX/DELT                              CK02390*
      IF(X) 206, 208, 207                                               CK02400*
  206 FLX(JSTR,2,K)=FLX(JSTR,2,K)-X                                     CK02410*
        GO TO 208                                                       CK02420*
  207 FLX(JSTR,1,K)=FLX(JSTR,1,K)+X                                     CK02430*
  208 CONTINUE                                                          CK02440*
C                                                                       CK02450*
C     ---LEAKAGE FROM RIVER CELLS OR OCEAN CELLS---                     CK02460*
      IF (NRIV.LE.0 .OR. IDR(I,J,K).LE.0) GO TO 213                     CK02470*
      ND=IDR(I,J,K)                                                     CK02480*
      IF(ISALT.LE.0 .OR. ISEA(I,J,K).LE.0) GO TO 209                    CK02490*
      IF((ISEA(I,J,K).EQ.2 .AND. T(I,J,KP1).GT.0.) .OR. PHIX.LE.RH(ND)) CK02500*
     1   GO TO 213                                                      CK02510*
  209 X=AMAX1(PHIX, RB(ND))                                             CK02520*
      Y=RC(ND)*(RH(ND)-X)*AX                                            CK02530*
      IF (IPRFLX.GT.0) WRITE (IDSK,9113) K,I,J, Y                       CK02540*
      IF(Y) 210, 213, 211                                               CK02550*
  210 FLX(JRIV,2,K)=FLX(JRIV,2,K)-Y                                     CK02560*
        GO TO 212                                                       CK02570*
  211 FLX(JRIV,1,K)=FLX(JRIV,1,K)+Y                                     CK02580*
  212 FL(I,J,K)=Y                                                       CK02590*
  213 CONTINUE                                                          CK02600*
C                                                                       CK02610*
C     ---DISCHARGE INTO DRAINS--                                        CK02620*
      IF (NDRAIN.LE.0 .OR. IDN(I,J,K).LE.0) GO TO 216                   CK02630*
      ND=IDN(I,J,K)                                                     CK02640*
      X=DRELV(ND)-PHIX                                                  CK02650*
      IF(X.GT.0.) X=0.                                                  CK02660*
      Y=DRCOF(ND)*X*AX                                                  CK02670*
      IF (IPRFLX.GT.0) WRITE (IDSK,9114) K,I,J, Y                       CK02680*
      FLX(JDRN,2,K)=FLX(JDRN,2,K)-Y                                     CK02690*
  215 GL(I,J,K)=Y                                                       CK02700*
  216 CONTINUE                                                          CK02710*
C                                                                       CK02720*
C     ---VERTICAL LEAKAGE ACROSS TOP OF LAYER---                        CK02730*
      IF(K.EQ.K0 .OR. T(I,J,KP1).LE.0. .OR. (S(I,J,K).LT.0. .AND.       CK02740*
     + S(I,J,KP1).LT.0.)) GO TO 186                                     CK02750*
```

```
      X=(SNGL(PHI(I,J,KP1))-PHIX)*TK(I,J,K)*AREA                        CK02760*
      IF(S(I,J,KP1).GE.0.) GO TO 184                                    CK02770*
        ICHD=1                                                          CK02780*
        IF(X.LT.0.) FLXN=FLXN-X                                         CK02790*
        IF(X.GT.0.) FLXP=FLXP+X                                         CK02800*
        IF(IPRFLX.EQ.0) GO TO 186                                       CK02810*
        IF(IPRT.GT.0) GO TO 182                                         CK02820*
          WRITE (IDSK,9111) KP1,I,J, K,I,J, X                           CK02830*
          IPRT=1                                                        CK02840*
          GO TO 186                                                     CK02850*
  182   WRITE (IDSK,9112) K,I,J, X                                      CK02860*
        GO TO 186                                                       CK02870*
  184 IF(X.LT.0.) FLX(JLKT,2,K)=FLX(JLKT,2,K)-X                         CK02880*
      IF(X.GT.0.) FLX(JLKT,1,K)=FLX(JLKT,1,K)+X                         CK02890*
  186 CONTINUE                                                          CK02900*
C                                                                       CK02910*
C     ---VERTICAL LEAKAGE ACROSS BOTTOM OF LAYER---                     CK02920*
      IF(K.EQ.1 .OR. T(I,J,KM1).LE.0. .OR. (S(I,J,K).LT.0. .AND.        CK02930*
     + S(I,J,KM1).LT.0.)) GO TO 192                                     CK02940*
      X=(SNGL(PHI(I,J,KM1))-PHIX)*TK(I,J,KM1)*AREA                      CK02950*
      IF(S(I,J,KM1).GE.0.) GO TO 190                                    CK02960*
        ICHD=1                                                          CK02970*
        IF(X.LT.0.) FLXN=FLXN-X                                         CK02980*
        IF(X.GT.0.) FLXP=FLXP+X                                         CK02990*
        IF(IPRFLX.EQ.0) GO TO 192                                       CK03000*
        IF(IPRT.GT.0) GO TO 188                                         CK03010*
          WRITE (IDSK,9111) KM1,I,J, K,I,J, X                           CK03020*
          IPRT=1                                                        CK03030*
          GO TO 192                                                     CK03040*
  188   WRITE (IDSK,9112) K,I,J, X                                      CK03050*
        GO TO 192                                                       CK03060*
  190 IF(X.LT.0.) FLX(JLKB,2,K)=FLX(JLKB,2,K)-X                         CK03070*
      IF(X.GT.0.) FLX(JLKB,1,K)=FLX(JLKB,1,K)+X                         CK03080*
  192 CONTINUE                                                          CK03090*
C                                                                       CK03100*
  218 IF(ICHD.EQ.0) GO TO 220                                           CK03110*
      FLX(JCHD,2,K)=FLX(JCHD,2,K)+FLXN                                  CK03120*
      FLX(JCHD,1,K)=FLX(JCHD,1,K)+FLXP                                  CK03130*
      EL(I,J,K)=FLXP-FLXN                                               CK03140*
C                                                                       CK03150*
  220 CONTINUE                                                          CK03160
C                                                                       CK03170*
      IF(IPRFLX.GT.0) REWIND IDSK                                       CK03180*
C                                                                       CK03190*
C....................................                                   CK03200*
C                                                                       CK03210*
C...SAVE FLUX VALUES ON DISK FILE.                                      CK03220*
      IF(ISVFLX.LE.0 .OR. (ISVFLX.EQ.1 .AND. IFINAL.NE.2)) GO TO 227    CK03230*
      IDSK=14                                                           CK03240*
      WRITE(6,9012) IDSK, KP, KT, SUM                                   CK03250*
C                                                                       CK03260*
      DO 224 K=1,K0                                                     CK03270*
  224 WRITE(IDSK) SETID(1), SETID(2), SUM, K, I0, J0,                   CK03280*
     1  ((EL(I,J,K), I=1,I0), J=1,J0)                                   CK03290*
C                                                                       CK03300*
```

```
      DO 225 K=1,K0                                                     CK03310*
  225 WRITE(IDSK) SETID(3), SETID(4), SUM, K, I0, J0,                   CK03320*
     1  ((FL(I,J,K), I=1,I0), J=1,J0)                                   CK03330*
C                                                                       CK03340*
      DO 226 K=1,K0                                                     CK03350*
  226 WRITE(IDSK) SETID(5), SETID(6), SUM, K, I0, J0,                   CK03360*
     1  ((GL(I,J,K), I=1,I0), J=1,J0)                                   CK03370*
  227 CONTINUE                                                          CK03380*
C                                                                       CK03390*
C...COMPUTE  TOTALS FOR FLOW RATES AND VOLUMES.                         CK03400*
C   NOTE THE MASS BALANCE IS COMPUTED AS                                CK03410*
C     (INFLOW+STORAGE DECREASE) - (OUTFLOW+STORAGE INCREASE) = RESIUDAL.CK03420*
      DO 230 K=1,K0                                                     CK03430*
      DO 230 J=1,2                                                      CK03440*
        JJ=J+2                                                          CK03450*
      DO 230 I=1,NFLX                                                   CK03460*
        X=FLX(I,J,K)                                                    CK03470*
        SUMFLX(J,K)=SUMFLX(J,K)+X                                       CK03480*
        Y=X*DELT                                                        CK03490*
        CFLX(I,J,K)=CFLX(I,J,K)+Y                                       CK03500*
        SUMFLX(JJ,K)=SUMFLX(JJ,K)+Y                                     CK03510*
      IF(I.GT.NFLX2) GO TO 230                                          CK03520*
        TFLX(I,J)=TFLX(I,J)+X                                           CK03530*
        CTFLX(I,J)=CTFLX(I,J)+Y                                         CK03540*
        SUMFLX(J,KK1)=SUMFLX(J,KK1)+X                                   CK03550*
        SUMFLX(JJ,KK1)=SUMFLX(JJ,KK1)+Y                                 CK03560*
  230 CONTINUE                                                          CK03570*
C                                                                       CK03580*
      RES1=SUMFLX(1,KK1)-SUMFLX(2,KK1)                                  CK03590*
      X=SUMFLX(1,KK1)+SUMFLX(2,KK1)                                     CK03600*
      PCT1=9999.                                                        CK03610*
      IF(X.NE.0.) PCT1=RES1*200./X                                      CK03620*
      RES2=SUMFLX(3,KK1)-SUMFLX(4,KK1)                                  CK03630*
      Y=SUMFLX(3,KK1)+SUMFLX(4,KK1)                                     CK03640*
      PCT2=9999.                                                        CK03650*
      IF(Y.NE.0.) PCT2=RES2*200./Y                                      CK03660*
      Y=TFLX(JSTR,1)+TFLX(JSTR,2)                                       CK03670*
      IF(X.EQ.0. .OR. ABS(Y*2./X).GE.STEADY) GO TO 232                  CK03680*
        IFINAL=2                                                        CK03690*
        ISTEDY=1                                                        CK03700*
        NPER=KP                                                         CK03710*
        NUMT=KT                                                         CK03720*
        WRITE(6,9100) KP, KT                                            CK03730*
  232 CONTINUE                                                          CK03740*
      GO TO 998                                                         CK03750*
C-----------------------------------------------------------------------CK03760*
C                                                                       CK03770*
C     ************************                                          CK03780
      ENTRY CWRITE                                                      CK03790
C     ************************                                          CK03800
C                                                                       CK03810
C...PRINT MASS BALANCE TABLES.                                          CK03820*
C                                                                       CK03830*
      DAYSIM=SUM/86400.                                                 CK03840*
      DAYPMP=SUMP/86400.                                                CK03850*
```

```
      DELSTP=DELT/86400.                                                  CK03860*
      WRITE(6,9000) KP, KT, DELSTP, DAYPMP, DAYSIM                        CK03870*
C                                                                         CK03880*
C...PRINT MASS BALANCE FOR INDIVIDUAL LAYERS.                             CK03890*
      IF(IBALAY.LE.0) GO TO 240                                           CK03900*
        JJ=-1                                                             CK03910*
      DO 239 K=1,K0                                                       CK03920*
        IF(SUMFLX(1,K).EQ.0. .AND. SUMFLX(2,K).EQ.0.) GO TO 239           CK03930*
        JJ=JJ+1                                                           CK03940*
        IF(JJ.GT.0 .AND. MOD(JJ,2).EQ.0) WRITE(6,9007)                    CK03950*
        WRITE(6,9001) K, KP, KT                                           CK03960*
        WRITE(6,9003)                                                     CK03970*
        WRITE(6,9004)                                                     CK03980*
        J=1                                                               CK03990*
      DO 238 I=1,NFLX                                                     CK04000*
        WRITE(6,9005) NAME(J), NAME(J+1), FLX(I,1,K), FLX(I,2,K),         CK04010*
     1   CFLX(I,1,K), CFLX(I,2,K)                                         CK04020*
        J=J+2                                                             CK04030*
  238 CONTINUE                                                            CK04040*
        WRITE(6,9004)                                                     CK04050*
        WRITE(6,9005) NAME(19), NAME(20), (SUMFLX(J,K), J=1,4)            CK04060*
        RES1L=SUMFLX(1,K)-SUMFLX(2,K)                                     CK04070*
        X=SUMFLX(1,K)+SUMFLX(2,K)                                         CK04080*
        PCT1L=9999.                                                       CK04090*
        IF(X.NE.0.) PCT1L=RES1L*200./X                                    CK04100*
        RES2L=SUMFLX(3,K)-SUMFLX(4,K)                                     CK04110*
        Y=SUMFLX(3,K)+SUMFLX(4,K)                                         CK04120*
        PCT2L=9999.                                                       CK04130*
        IF(Y.NE.0.) PCT2L=RES2L*200./Y                                    CK04140*
        WRITE(6,9006) RES1L, RES2L, PCT1L, PCT2L                          CK04150*
  239 CONTINUE                                                            CK04160*
        WRITE(6,9007)                                                     CK04170*
  240 CONTINUE                                                            CK04180*
C                                                                         CK04190*
C...PRINT COMBINED MASS BALANCE (TOTALS FOR ALL LAYERS).                  CK04200*
      WRITE(6,9002) KP, KT                                                CK04210*
      WRITE(6,9003)                                                       CK04220*
      WRITE(6,9004)                                                       CK04230*
        J=1                                                               CK04240*
      DO 244 I=1,NFLX2                                                    CK04250*
        WRITE(6,9005) NAME(J), NAME(J+1), TFLX(I,1), TFLX(I,2),           CK04260*
     1     CTFLX(I,1), CTFLX(I,2)                                         CK04270*
        J=J+2                                                             CK04280*
  244 CONTINUE                                                            CK04290*
      WRITE(6,9004)                                                       CK04300*
      WRITE(6,9005) NAME(19), NAME(20), (SUMFLX(J,KK1), J=1,4)            CK04310*
      WRITE(6,9006) RES1, RES2, PCT1, PCT2                                CK04320*
C                                                                         CK04330*
C...PRINT VERTICAL FLUX ACROSS TOPS OF MODEL LAYERS.                      CK04340*
      IF(K0.EQ.1) GO TO 251                                               CK04350*
        WRITE(6,9115) KP, KT                                              CK04360*
C                                                                         CK04370*
      DO 250 K=1,K1                                                       CK04380*
        KP1=K+1                                                           CK04390*
        WRITE(6,9116) K, FLX(JLKT,2,K), FLX(JLKT,1,K)                     CK04400*
```

```
  250 CONTINUE                                                          CK04410*
  251 CONTINUE                                                          CK04420*
C                                                                       CK04430*
C...PRINT FLUX VALUES COMPUTED FOR THIS TIME STEP.                      CK04440*
      IF(IPFLX.LE.0) GO TO 264                                          CK04450*
        IDSK=13                                                         CK04460*
  260 READ(IDSK,8001,END=262) X, AFLX                                   CK04470*
        WRITE(6,8001) X, AFLX                                           CK04480*
        GO TO 260                                                       CK04490*
  262 REWIND IDSK                                                       CK04500*
  264 CONTINUE                                                          CK04510*
C                                                                       CK04520*
      GO TO 998                                                         CK04530*
C-----------------------------------------------------------------------CK04540*
C                                                                       CK04550*
C...ERROR MESSAGES...                                                   CK04560*
  901 WRITE(6,9901) NSUBAL, MXSUB                                       CK04570*
  997 WRITE(6,9997)                                                     CK04580*
      STOP                                                              CK04590*
C                                                                       CK04600*
  998 RETURN                                                            CK04610*
C                                                                       CK04620*
C-----------------------------------------------------------------------CK04630*
C                                                                       CK04640*
 8000 FORMAT(4I5)                                                       CK04650*
 8001 FORMAT(A1,33A4)                                                   CK04660*
 9000 FORMAT(1H1,5X, 59(1H-) / 6X,   35HMASS BALANCE FOR STRESS PERIOD NCK04670*
     1O. ,I4, 16H, TIME STEP NO. ,I4, / 6X, 59(1H-) / 1H-,              CK04680*
     2  8X, 29HDURATION OF CURRENT TIME STEP,6X, 3H = ,F12.5,5H DAYS/   CK04690*
     3  9X,33HDURATION OF CURRENT STRESS PERIOD,2X,3H = ,F12.5,5H DAYS/ CK04700*
     4  9X, 21HTOTAL SIMULATION TIME,14X, 3H = ,F12.5, 5H DAYS)         CK04710*
 9001 FORMAT(/1H-,5X, 27HMASS BALANCE FOR LAYER NO. ,I3, 20H, STRESS PERCK04720*
     1IOD NO. ,I4, 16H, TIME STEP NO. ,I4 / 6X, 74(1H-) )               CK04730*
 9002 FORMAT(/1H-,5X, 29HMASS BALANCE FOR ENTIRE MODEL, 20H, STRESS PERICK04740*
     1OD NO. ,I4, 16H, TIME STEP NO. ,I4 / 6X, 73(1H-) )                CK04750*
 9003 FORMAT(1H-,8X, 8HQUANTITY,9X, 8HRATE  IN,7X, 8HRATE OUT,6X,       CK04760*
     1 14HCUMUL. VOL. IN,6X, 15HCUMUL. VOL. OUT)                        CK04770*
 9004 FORMAT(6X,15(1H-),3X, 12(1H-),3X, 12(1H-),3X, 17(1H-),3X, 17(1H-))CK04780*
 9005 FORMAT(6X, A8,A7, 2F15.3, 2F20.0)                                 CK04790*
 9006 FORMAT(6X, 8HIN - OUT,14X,   F15.3,15X, F20.0 /                   CK04800*
     1       6X,10HPCT. DIFF.,13X, F13.2,18X, F20.2)                    CK04810*
 9007 FORMAT(1H1)                                                       CK04820*
 9010 FORMAT( 1H-, 'DATA FROM RECORD 8 IN GROUP I DATA: MODEL SUBAREAS FCK04830*
     1OR MASS BALANCE COMPUTATIONS' / 1X, 80(1H-) /                     CK04840*
     2 1H0,22X, 'ROWS        COLUMNS' /                                 CK04850*
     3 11X, 'SUBAREA   TOP  BOT.   LEFT  RIGHT' /                       CK04860*
     4 11X, '-------   ---  ----   ----  -----' )                       CK04870*
 9011 FORMAT(12X,I3, 4X,2I5, 1X,2I6)                                    CK04880*
 9012 FORMAT( '-*** COMPUTED FLUX RATES SAVED ON FORTRAN FILE NO. ',I2, CK04890*
     1 ' FOR STRESS PERIOD = ',I4, ', TIME STEP = ',I4, ', SIMULATION TICK04900*
     2ME = ',F14.2, ' ***')                                             CK04910*
 9020 FORMAT(' ***ERROR IN SUBR. CHECKI*** INCORRECT COORDINATE VALUE.')CK04920*
 9100 FORMAT(/ '-*** STEADY-STATE CONDITION IN STRESS PERIOD ',I4,      CK04930*
     1   ' TIME STEP ',I4, 4H ***//)                                    CK04940*
 9110 FORMAT(1H1,4X, 32HCOMPUTED FLUX FOR STRESS PERIOD ,I3, 12H, TIME SCK04950*
```

```
     1TEP ,I3, 21H, SIMULATION TIME OF ,F12.1 /5X,83(1H-)/            CK04960*
     2 1H0,6X, 46H(NEGATIVE VALUES INDICATE FLUX OUT OF AQUIFER) /     CK04970*
     3 1H-,5X, 44HFLUX BETWEEN AQUIFER AND CONSTANT-HEAD CELLS,5X,     CK04980*
     4 36HFLUX BETWEEN AQUIFER AND RIVER CELLS,9X, 16HFLUX INTO DRAINS /CK04990*
     4 6X, 44(1H.), 5X,36(1H.), 5X,25(1H.) /                           CK05000*
     5 6X, 41HCONSTANT-HEAD CELL  AQUIFER CELL     FLUX, 13X,          CK05010*
     6 21HAQUIFER CELL     FLUX,15X, 21HAQUIFER CELL     FLUX /        CK05020*
     7 3X, 2(6X, 11H(K,  I,  J)), 3X, 11H(FT**3/SEC), 9X,              CK05030*
     8 11H(K,  I,  J), 3X, 11H(FT**3/SEC) ,                            CK05040*
     8 11X, 11H(K,  I,  J),3X, 11H(FT**3/SEC) /                        CK05050*
     9 6X, 18(1H-), 2X, 12(1H-), 2X, 11(1H-), 9X, 12(1H-), 2X, 11(1H-), CK05060*
     X 11X,12(1H-), 2X, 11(1H-) /)                                     CK05070*
 9111 FORMAT(8X, 2(I3, 1H,), I3,6X, 2(I3,1H,), I3,4X, 1P,E11.4)        CK05080*
 9112 FORMAT(25X, 2(I3, 1H,), I3,4X, 1P,E11.4)                         CK05090*
 9113 FORMAT(59X, 2(I3, 1H,), I3,4X, 1P,E11.4)                         CK05100*
 9114 FORMAT(94X, 2(I3, 1H,), I3,4X, 1P,E11.4)                         CK05110*
 9115 FORMAT(/1H-,5X, 41HCOMPUTED FLUX ACROSS TOPS OF MODEL LAYERS,    CK05120*
     1 20H, STRESS PERIOD NO. ,I4, 16H, TIME STEP NO. ,I4/             CK05130*
     1 6X, 85(1H-) //                                                  CK05140*
     1 6X, '             UPWARD           DOWNWARD'/                   CK05150*
     2 6X, 'LAYER         FLUX              FLUX'/                     CK05160*
     3 6X, '-----   --------------   --------------')                  CK05170*
 9116 FORMAT(6X,I4, 2(4X, 1P,E13.6))                                   CK05180*
 9901 FORMAT('-***ERROR IN SUBR. CHECKI*** NO. MASS BAL. SUBAREAS = ', CK05190*
     1 I3, '  EXCEEDS ARRAY SIZE OF ',I3)                              CK05200*
 9997 FORMAT(  '-*** JOB EXECUTION TERMINATED ***'//)                  CK05210*
      END                                                              CK05220
```

```
      SUBROUTINE COEF(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACTCF00010
     1,PERM,BOTTOM)                                                     CF00020*
C     ------------------------------------------------------------------CF00030
C     COMPUTE COEFFICIENTS                                              CF00040
C     ------------------------------------------------------------------CF00050
C                                                                       CF00060
C     SPECIFICATIONS:                                                   CF00070
      REAL*8  PHI                                                       CF00080
C                                                                       CF00090
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0CF00100
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,CF00110
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(1,3,K0), PERM(IP,JP),  CF00120*
     3 BOTTOM(IP,JP)                                                    CF00130*
C                                                                       CF00140
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,      NPER,KTH,ITMAX,LENGTH,KP,NCF00150*
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCCF00160
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN   CF00170*
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC   CF00180*
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD       CF00190*
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY                               CF00200*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1      CF00210*
     1, ERMULT, STEADY, DELT1                                           CF00220*
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)       CF00230*
      GO TO 100                                                         CF00240*
C     ..................................................................CF00250
C     **********************                                            CF00260
      ENTRY TRANS                                                       CF00270*
C     **********************                                            CF00280
C                                                                       CF00290*
C...COMPUTE TRANSMISSIVITY FOR WATER-TABLE AQUIFER IN LAYER NO.         CF00300*
C   LWTABL OF MODEL.                                                    CF00310*
C                                                                       CF00320*
        KK=LWTABL                                                       CF00330*
C                                                                       CF00340*
      DO 10 I=2,I1                                                      CF00350
      DO 10 J=2,J1                                                      CF00360
      IF (PERM(I,J).LE.0.) GO TO 8                                      CF00370*
      T(I,J,KK)=PERM(I,J)*(SNGL(PHI(I,J,KK))-BOTTOM(I,J))               CF00380*
      IF (T(I,J,KK).GT.0.) GO TO 9                                      CF00390*
      IF (WELL(I,J,KK).LT.0.) WRITE (6,60) I,J,KK                       CF00400*
      IF (WELL(I,J,KK).GE.0.) WRITE (6,70) I,J,KK                       CF00410*
      PHI(I,J,KK)=-99.                                                  CF00420*
      PERM(I,J)=0.                                                      CF00430
      IF(KK.GT.1) TK(I,J,KK-1)=0.                                       CF00440*
    8 T(I,J,KK)=0.                                                      CF00450*
      GO TO 10                                                          CF00460*
    9 IF(IEQN.EQ.ICHK(11)) T(I,J,KK)=PERM(I,J)                          CF00470*
   10 CONTINUE                                                          CF00480
C                                                                       CF00490*
      N1=LWTABL                                                         CF00500*
      N2=N1                                                             CF00510*
      N3=LWTABL-1                                                       CF00520*
      N4=N3                                                             CF00530*
      GO TO 20                                                          CF00540
C                                                                       CF00550*
```

```
C                                                                       CF00560*
C     *************************                                         CF00570
      ENTRY TCOF                                                        CF00580
C     *************************                                         CF00590
C                                                                       CF00600*
C...COMPUTE HARMONIC MEANS FOR TRANSMISSIVITIES IN ADJACENT CELLS.  NOTECF00610*
C   THAT THE ARRAYS TR, TC, AND TK WERE SET TO ZERO IN SUBR. DATAI AT   CF00620*
C   THE START OF SIMULATION.                                            CF00630*
      N1=1                                                              CF00640
      N2=0                                                              CF00650*
      N3=1                                                              CF00660*
      IF(NTR.EQ.1) N2=LWTABL-1                                          CF00670*
      IF(NTR.EQ.2) N2=K0                                                CF00680*
      N4=N2-1                                                           CF00690*
      IF(N2.LE.0) GO TO 100                                             CF00700*
C                                                                       CF00710*
C   ---COMPUTE HARMONIC MEANS BETWEEN CELLS IN THE SAME LAYER---        CF00720*
   20 DO 40 K=N1,N2                                                     CF00730
        F1=FACT(1,1,K)                                                  CF00740*
        F2=FACT(1,2,K)                                                  CF00750*
      DO 40 I=2,I1                                                      CF00760*
        IP1=I+1                                                         CF00770*
        Z1=DELY(I)                                                      CF00780*
        Z2=DELY(IP1)                                                    CF00790*
      DO 40 J=2,J1                                                      CF00800*
      X1=0.                                                             CF00810*
      X2=0.                                                             CF00820*
      T1=T(I,J,K)                                                       CF00830*
      IF(T1.LE.0.) GO TO 33                                             CF00840*
      JP1=J+1                                                           CF00850*
      T2=T(I,JP1,K)                                                     CF00860*
      IF(T2.GT.0.) X1=2.*T1*T2*F1/(T1*DELX(JP1)+T2*DELX(J))             CF00870*
      T2=T(IP1,J,K)                                                     CF00880*
      IF(T2.GT.0.) X2=2.*T1*T2*F2/(T1*Z2 + T2*Z1)                       CF00890*
   33 TR(I,J,K)=X1                                                      CF00900*
      TC(I,J,K)=X2                                                      CF00910*
   40 CONTINUE                                                          CF00920
C                                                                       CF00930*
C   ---COMPUTE HARMONIC MEANS BETWEEN CELLS IN DIFFERENT LAYERS---      CF00940*
      IF (N4.LE.0 .OR. ITK.EQ.ICHK(10)) GO TO 100                       CF00950*
      DO 50 K=N3,N4                                                     CF00960*
        KP1=K+1                                                         CF00970*
        F1=FACT(1,3,K)                                                  CF00980*
        F2=FACT(1,3,KP1)                                                CF00990*
        Z1=DELZ(K)                                                      CF01000*
        Z2=DELZ(KP1)                                                    CF01010*
      DO 50 I=2,I1                                                      CF01020
      DO 50 J=2,J1                                                      CF01030
      X1=0.                                                             CF01040*
      T1=T(I,J,K)*F1                                                    CF01050*
      IF(T1.LE.0.) GO TO 49                                             CF01060*
      T2=T(I,J,KP1)*F2                                                  CF01070*
      IF(KP1.NE.LWTABL) GO TO 48                                        CF01080*
        T2=PERM(I,J)*F2                                                 CF01090*
        Z2=SNGL(PHI(I,J,LWTABL))-BOTTOM(I,J)                            CF01100*
```

```
   48 IF(T2.GT.0.) X1=2.*T1*T2/(T1*Z2+T2*Z1)                          CF01110*
   49 TK(I,J,K)=X1                                                    CF01120*
   50 CONTINUE                                                        CF01130
C                                                                     CF01140*
  100 RETURN                                                          CF01150*
C                                                                     CF01160*
C                                                                     CF01170*
   60 FORMAT ('-',20('*'),'WELL',2I3,' IN LAYER',I3,' GOES DRY',20('*'))CF01180
   70 FORMAT ('-',20('*'),'CELL',2I3,' IN LAYER',I3,' GOES DRY',20('*'))CF01190*
      END                                                             CF01200
```

```
      SUBROUTINE DATAI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACDA00010
     1T,PERM,BOTTOM,QRE,LAYRCH,V,IDR,RH,RC,RB,IDN,DRCOF,DRELV,ISEA)      DA00020*
C     -----------------------------------------------------------------DA00030
C     READ PARAMETERS FOR DATA GROUPS I, II, IV, AND V.                DA00040*
C     -----------------------------------------------------------------DA00050
C                                                                      DA00060
C     SPECIFICATIONS:                                                  DA00070
      REAL*8  PHI, TFLX, CFLX, CTFLX, SUMFLX                           DA00080*
      REAL*8  XLABEL,YLABEL,TITLE,XN1,MESUR                            DA00090
C                                                                      DA00100
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0DA00110
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,DA00120
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(1,3,K0), PERM(IP,JP),  DA00130*
     3BOTTOM(IP,JP), QRE(IQ,JQ), V(I0,J0,K0), IDR(IA1,JA1,KA1), RH(1),  DA00140*
     4 RC(1), RB(1), IDN(IA2,JA2,KA2), DRCOF(1), DRELV(1), ISEA(IA3,JA3,DA00150*
     5KA3),LAYRCH(IQ,JQ), SETNAM(40), SETID(2), DBASID(4)               DA00160*
C                                                                      DA00170*
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,      NPER,KTH,ITMAX,LENGTH,KP,NDA00180*
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCDA00190
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN   DA00200*
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC   DA00210*
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD       DA00220*
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY                               DA00230*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1      DA00240*
     1, ERMULT, STEADY, DELT1                                           DA00250*
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)       DA00260*
      COMMON /CK/ SUMFLX(4,41), TFLX(9,2), CFLX(9,2,40), CTFLX(9,2),    DA00270*
     1  FLX(9,2,40)                                                     DA00280*
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANKDA00290
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),DA00300
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                          DA00310
C                                                                      DA00320*
      DATA SETNAM/4HSTRT,1H , 4HS   ,1H , 4HT   ,1H , 4HTXTY,1H ,       DA00330*
     1  4HTK  ,1H , 4HPERM,1H , 4HBOT ,1H , 4HQRE ,1H , 4HDELX,1H ,     DA00340*
     2  4HDELY,1H , 4HDELZ,1H , 4HWELL,1H , 4HIDN ,1H , 4HDRCF,1H ,     DA00350*
     3  4HDREL,1H , 4HIDR ,1H , 4HRC  ,1H , 4HRB  ,1H , 4HRH  ,1H ,     DA00360*
     4  4HLAYR,2HCH /                                                   DA00370*
      DATA DBASID /4HOPEN,1H , 4HCLOS,1HE/                              DA00380*
      DATA IDKR1,IBLANK/4HHEDR, 1H /                                    DA00390*
      GO TO 998                                                         DA00400*
C     ..................................................................DA00410
C     **********************                                            DA00420
      ENTRY DATAIN                                                      DA00430
C     **********************                                            DA00440
C                                                                      DA00450
      KPM1=0                                                            DA00460*
C                                                                      DA00470*
C...READ RECORD 9 OF GROUP I DATA...                                   DA00480*
      READ (5,330) NPER,KTH,LENGTH,ERR,ERMULT,WMAX1,BETA1,STEADY,ITRHED,DA00490*
     1  IPHED                                                           DA00500*
        IF(NPER.LE.0) NPER=1                                            DA00510*
        IF(KTH.LE.0) KTH=1                                              DA00520*
        IF(ERR.LE.0.) ERR=.1                                            DA00530*
        IF(ERMULT.LE.0.) ERMULT=1.                                      DA00540*
        IF(LENGTH.LE.0) LENGTH=5                                        DA00550*
```

```
        IF(BETA1.LE.0.) BETA1=1.
      WRITE (6,340) NPER,KTH,LENGTH,ERR,ERMULT,WMAX1,BETA1,STEADY,
     1  ITRHED,IPHED
C
C...READ RECORDS 10 AND 11 OF GROUP I DATA...
      READ (5,460) XSCALE,YSCALE,DINCH,FACT1,FACT2,MESUR
      IF(YSCALE.LT.0.) READ (5,461) (LEVEL1(K),K=1,40), (LEVEL2(K),K=1,
     1 40)
C
      DO 2 K=1,KO
        J=K
        IF(YSCALE.LT.0. .AND. LEVEL1(K).LE.0) J=0
        LEVEL1(K)=J
        J=K
        IF(YSCALE.LT.0. .AND. LEVEL2(K).LE.0) J=0
        LEVEL2(K)=J
    2 CONTINUE
C
      IF(FACT1.EQ.0.) FACT1=1.
      IF(FACT2.EQ.0.) FACT2=1.
C
      WRITE (6,470) XSCALE,YSCALE,DINCH,FACT1,FACT2,MESUR
      IF(IDRAW.EQ.ICHK(1) .OR. XSCALE.GT.0.) WRITE (6,462) (LEVEL1(K),
     1   K=1,KO)
      IF(IHEAD.EQ.ICHK(2) .OR. XSCALE.GT.0.) WRITE (6,463) (LEVEL2(K),
     1   K=1,KO)
      YSCALE=ABS(YSCALE)
C
C...INITIALIZE MASS BALANCE PARAMETERS...
      SUM=0.
      SUMP=0.
C
      DO 8 I=1,9
      DO 8 J=1,2
        CTFLX(I,J)=0.D0
      DO 8 K=1,KO
        CFLX(I,J,K)=0.D0
    8 CONTINUE
C
C...READ RECORDS 1 - 5 OF GROUP II DATA...
      IF (IPU1.NE.ICHK(8)) GO TO 20
        IDK1=IBLANK
C
C     ---READ TIMES OF PREVIOUS SIMULATION FROM FORMATTED FILE---
      IDSK=3
      READ(IDSK,450) SUM, SUMP
      WRITE(6,430) IDSK, SUM, SUMP
C
C     ---READ MASS BALANCE PARAMETERS FROM FORMATTED FILE---
      DO 4 K=1,KO
      DO 4 J=1,2
    4 READ(IDSK,450) (CFLX(I,J,K), I=1,9)
C
C     ---READ INITIAL HEAD VALUES FROM FORMATTED FILE---
      DO 10 K=1,KO
```

```
      DO 10 I=1,I0                                                      DA01110
   10 READ (IDSK,360) (PHI(I,J,K),J=1,J0)                               DA01120*
      IF(IDSK.NE.5) REWIND IDSK                                         DA01130*
      GO TO 30                                                          DA01140
C                                                                       DA01150
C     ---READ INITIAL HEAD AND MASS BALANCE PARAMETERS FROM UNFORMATTED DA01160*
C        DISK FILE---                                                   DA01170*
   20 IF(IDK1.EQ.IBLANK) GO TO 41                                       DA01180*
      IPU1=IBLANK                                                       DA01190*
      IDSK=4                                                            DA01200*
      DO 24 K=1,K0                                                      DA01210*
   24 READ(IDSK) ((PHI(I,J,K), I=1,I0),J=1,J0)                          DA01220*
      IF (IDK1.EQ.ICHK(4))                                              DA01230*
     1   READ(IDSK) SUM, SUMP, (((CFLX(I,J,K), I=1,9), J=1,2), K=1,K0)  DA01240*
      REWIND IDSK                                                       DA01250*
      WRITE(6,431) IDSK, SUM, SUMP                                      DA01260*
C                                                                       DA01270*
   30 IF(IPHED.LE.0) GO TO 41                                           DA01280*
      DO 40 K=1,K0                                                      DA01290
      WRITE (6,440) K, SUM                                              DA01300*
      DO 40 I=1,I0                                                      DA01310
   40 WRITE (6,350) I,(PHI(I,J,K),J=1,J0)                               DA01320
C                                                                       DA01330
   41 CONTINUE                                                          DA01340*
C                                                                       DA01350*
C...INITIALIZE THE ARRAY CTFLX FOR SUBR. CHECKI.                        DA01360*
      DO 52 K=1,K0                                                      DA01370*
      DO 52 J=1,2                                                       DA01380*
      DO 52 I=1,9                                                       DA01390*
   52 CTFLX(I,J)=CTFLX(I,J)+CFLX(I,J,K)                                 DA01400*
C                                                                       DA01410*
C                                                                       DA01420*
C...READ GROUP V DATA...                                                DA01430*
C                                                                       DA01440*
      DO 60 K=1,K0                                                      DA01450
      DO 60 J=1,J0                                                      DA01460
      DO 60 I=1,I0                                                      DA01470
      TR(I,J,K)=0.                                                      DA01480
      TC(I,J,K)=0.                                                      DA01490
      IF (K.NE.K0) TK(I,J,K)=0.                                         DA01500
   60 CONTINUE                                                          DA01510
C                                                                       DA01520*
      CALL GWDBAS(1, DBASID(1), V, 0, 0, K0, I0, J0, 0, 0, 0.)          DA01530*
      CALL GWDBAS(0, SETNAM(1), STRT,  3, 1, K0, I0, J0, 0, 0, 0.)      DA01540*
      CALL GWDBAS(0, SETNAM(3), S,     3, 1, K0, I0, J0, 0, 0, 0.)      DA01550*
      CALL GWDBAS(0, SETNAM(5), T,     3, 1, K0, I0, J0, 0, 0, 0.)      DA01560*
      CALL GWDBAS(0, SETNAM(7), FACT,  4, 1, K0, 1, 3, 0, 0, 0.)        DA01570*
      IF (ITK.EQ.ICHK(10)) CALL GWDBAS(0, SETNAM(9), TK, 3, 1, K1, I0,  DA01580*
     1     J0, 0, 0, 0.)                                                DA01590*
      IF (IWATER.NE.ICHK(6)) GO TO 62                                   DA01600*
        CALL GWDBAS(0, SETNAM(11), PERM, 2, 1, 1, I0, J0, 0, 0, 0.)     DA01610*
        CALL GWDBAS(0, SETNAM(13), BOTTOM, 2, 1, 1, I0, J0, 0, 0, 0.)   DA01620*
   62 IF (IQRE.EQ.ICHK(7)) CALL GWDBAS(0, SETNAM(15), QRE, 2, 1, 1, I0, DA01630*
     1     J0, 0, 0, 0.)                                                DA01640*
      IF(LRECH.LT.0) CALL GWDBAS (0, SETNAM(39), V, 2, 1, 1, I0, J0,    DA01650*
```

```
     1  0, 0, 0.)                                                        DA01660*
      CALL GWDBAS(0, SETNAM(17), DELX, 1, 0, 0, 0, J0, 0, 0, 0.)        DA01670*
      CALL GWDBAS(0, SETNAM(19), DELY, 1, 0, 0, 0, I0, 0, 0, 0.)        DA01680*
      CALL GWDBAS(0, SETNAM(21), DELZ, 1, 0, 0, 0, K0, 0, 0, 0.)        DA01690*
      CALL GWDBAS(0, SETNAM(23), WELL, 3, 1, K0, I0, J0, 0, 0, 0.)      DA01700*
C                                                                       DA01710*
C...RESET VALUES IN DATA ARRAYS AND COUNT CONSTANT-HEAD CELLS.          DA01720*
        N=0                                                             DA01730*
        NN=0                                                            DA01740*
        IF(IDK1.EQ.ICHK(4) .OR. IPU1.EQ.ICHK(8)) NN=1                   DA01750*
        IF(IDK1.EQ.IDKR1)  NN=2                                         DA01760*
      DO 150 K=1,K0                                                     DA01770
        KM1=K-1                                                         DA01780*
        IF(FACT(1,1,K).LE.0.) FACT(1,1,K)=1.                            DA01790*
        IF(FACT(1,2,K).LE.0.) FACT(1,2,K)=1.                            DA01800*
        IF(FACT(1,3,K).LE.0.) FACT(1,3,K)=1.                            DA01810*
      DO 150 J=1,J0                                                     DA01820
      DO 150 I=1,I0                                                     DA01830
      IF (NN.EQ.0) PHI(I,J,K)=STRT(I,J,K)                               DA01840*
      IF (NN.EQ.2) STRT(I,J,K)=PHI(I,J,K)                               DA01850*
      IF (I.GT.1 .AND. I.LT.I0 .AND. J.GT.1 .AND. J.LT.J0) GO TO 146    DA01860*
        T(I,J,K)=0.                                                     DA01870*
        IF (K.EQ.LWTABL) PERM(I,J)=0.                                   DA01880*
        GO TO 147                                                       DA01890*
  146 IF (K.EQ.LWTABL) T(I,J,K)=PERM(I,J)*(SNGL(PHI(I,J,K))-BOTTOM(I,J))DA01900*
      IF (T(I,J,K).LT.0.) T(I,J,K)=0.                                   DA01910*
      IF (T(I,J,K).GT.0. .AND. S(I,J,K).LT.0.) N=N+1                    DA01920*
  147 IF (K.GT.1 .AND. (T(I,J,K).LE.0. .OR. T(I,J,KM1).LE.0.))          DA01930*
     1   TK(I,J,KM1)=0.                                                 DA01940*
  150 CONTINUE                                                          DA01950
C                                                                       DA01960*
      WRITE (6,391) N, NCH                                              DA01970*
      IF (N.LE.NCH) GO TO 151                                           DA01980*
        WRITE (6,392)                                                   DA01990*
        GO TO 999                                                       DA02000*
  151 NCH=N                                                             DA02010*
C                                                                       DA02020*
C...DIVIDE CONSTANT FLUX VALUES BY AREA OR VOLUME OF EACH CELL.         DA02030*
        DZ=1.                                                           DA02040*
      DO 65 K=1,K0                                                      DA02050*
        IF(IEQN.EQ.ICHK(11)) DZ=DELZ(K)                                 DA02060*
      DO 65 J=1,J0                                                      DA02070*
        DX=DELX(J)*DZ                                                   DA02080*
      DO 65 I=1,I0                                                      DA02090*
   65 WELL(I,J,K)=WELL(I,J,K)/(DELY(I)*DX)                              DA02100*
C                                                                       DA02110*
C...DETERMINE LAYER TO BE RECHARGED AND, IF IEQN='EQN3', DIVIDE         DA02120*
C   RECHARGE VALUES BY DELZ.                                            DA02130*
        LAYRCH(1,1)=0                                                   DA02140*
        QRE(1,1)=0.                                                     DA02150*
      IF(IQRE.NE.ICHK(7)) GO TO 82                                      DA02160*
        K=K0                                                            DA02170*
        IF(LWTABL.GT.0) K=LWTABL                                        DA02180*
C                                                                       DA02190*
      IF(LRECH.GT.0) GO TO 74                                           DA02200*
```

```
      DO 72 J=1,J0                                                      DA02210*
      DO 72 I=1,I0                                                      DA02220*
        IF(LRECH.EQ.0) LAYRCH(I,J)=K                                    DA02230*
        IF(LRECH.LT.0) LAYRCH(I,J)=V(I,J,1)                             DA02240*
   72 CONTINUE                                                          DA02250*
        GO TO 78                                                        DA02260*
C                                                                       DA02270*
   74 DO 76 J=1,J0                                                      DA02280*
      DO 76 I=1,I0                                                      DA02290*
        LAYRCH(I,J)=0                                                   DA02300*
        K=K0                                                            DA02310*
      DO 75 KK=LRECH,K0                                                 DA02320*
        IF(T(I,J,K).LE.0.) GO TO 75                                     DA02330*
        IF(S(I,J,K).GE.0.) LAYRCH(I,J)=K                                DA02340*
        GO TO 76                                                        DA02350*
   75 K=K-1                                                             DA02360*
   76 CONTINUE                                                          DA02370*
C                                                                       DA02380*
   78 IF(IEQN.NE.ICHK(11)) GO TO 82                                     DA02390*
      DO 80 J=1,J0                                                      DA02400*
      DO 80 I=1,I0                                                      DA02410*
        K=LAYRCH(I,J)                                                   DA02420*
        IF(K.GT.0) QRE(I,J)=QRE(I,J)/DELZ(K)                            DA02430*
   80 CONTINUE                                                          DA02440*
   82 CONTINUE                                                          DA02450*
C                                                                       DA02460*
C                                                                       DA02470*
C...READ GROUP V DATA FOR DRAIN CELLS.                                  DA02480*
      IF(NDRAIN.LE.0) GO TO 550                                         DA02490*
C                                                                       DA02500*
C     ---READ THE ARRAY IDN---                                          DA02510*
      CALL GWDBAS(0, SETNAM(25), V, 3, 1, K0, I0, J0, 0, 0, 0.)         DA02520*
        N=0                                                             DA02530*
      DO 490 K=1,K0                                                     DA02540*
      DO 490 I=1,I0                                                     DA02550*
      DO 490 J=1,J0                                                     DA02560*
        NN=ABS(V(I,J,K))                                                DA02570*
        IF(NN.EQ.0) GO TO 489                                           DA02580*
          N=N+1                                                         DA02590*
          NN=N                                                          DA02600*
  489   IDN(I,J,K)=NN                                                   DA02610*
  490 CONTINUE                                                          DA02620*
C                                                                       DA02630*
      WRITE(6,394) N, NDRAIN                                            DA02640*
      IF(N.LE.NDRAIN) GO TO 491                                         DA02650*
        WRITE(6,392)                                                    DA02660*
        GO TO 999                                                       DA02670*
  491 NDRAIN=N                                                          DA02680*
C                                                                       DA02690*
C     ---READ THE ARRAY DRCOF---                                        DA02700*
      CALL GWDBAS(0, SETNAM(27), V, 3, 1, K0, I0, J0, 0, 0, 0.)         DA02710*
C                                                                       DA02720*
        DZ=1.                                                           DA02730*
      DO 505 K=1,K0                                                     DA02740*
        IF (IEQN.EQ.ICHK(11)) DZ=DELZ(K)                                DA02750*
```

```
      DO 505 I=1,I0
        DY=DELY(I)*DZ
      DO 505 J=1,J0
      NN=IDN(I,J,K)
      IF(NN.GT.0) DRCOF(NN)=V(I,J,K)/(DELX(J)*DY)
  505 CONTINUE
C
C     ---READ THE ARRAY DRELV---
      CALL GWDBAS (0, SETNAM(29), V, 3, 1, K0, I0, J0, 0, 0, 0.)
C
      DO 510 K=1,K0
      DO 510 I=1,I0
      DO 510 J=1,J0
        NN=IDN(I,J,K)
        IF(NN.GT.0) DRELV(NN)=V(I,J,K)
  510 CONTINUE
C
C...READ GROUP V DATA FOR LEAKY RIVER CELLS, AND INITIALIZE ISEA FOR
C   THE SEAWATER INTRUSION PROBLEM (ISALT=1).
  550 IF(NRIV.LE.0) GO TO 700
C
C     ---READ THE ARRAY IDR---
      CALL GWDBAS(0,SETNAM(31), V, 3, 1, K0, I0, J0, 0, 0, 0.)
C
C     ---INITIALIZE THE ARRAY ISEA IF ISALT.GT.0---
      IF(ISALT.LE.0) GO TO 571
      DO 570 K=1,K0
      DO 570 J=1,J0
      DO 570 I=1,I0
  570 ISEA(I,J,K)=0
  571 CONTINUE
C
C     ---RESET THE ARRAYS IDR AND ISEA---
        N=0
      DO 580 K=1,K0
      DO 580 I=1,I0
      DO 580 J=1,J0
        NN=ABS(V(I,J,K))
        IF(NN.EQ.0) GO TO 579
          IF (ISALT.GT.0 .AND. NN.GT.1) ISEA(I,J,K)=2
          N=N+1
          NN=N
  579   IDR(I,J,K)=NN
  580 CONTINUE
C
      WRITE(6,393) N, NRIV
      IF(N.LE.NRIV) GO TO 581
        WRITE(6,392)
        GO TO 999
  581 NRIV=N
C
C     ---READ THE ARRAY RC---
      CALL GWDBAS(0, SETNAM(33), V, 3, 1, K0, I0, J0, 0, 0, 0.)
C
        DZ=1.
```

```
      DO 605 K=1,K0                                                     DA03310*
        IF(IEQN.EQ.ICHK(11)) DZ=DELZ(K)                                 DA03320*
      DO 605 I=1,I0                                                     DA03330*
        DY=DELY(I)*DZ                                                   DA03340*
      DO 605 J=1,J0                                                     DA03350*
        NN=IDR(I,J,K)                                                   DA03360*
        IF(NN.GT.0) RC(NN)=V(I,J,K)/(DELX(J)*DY)                        DA03370*
  605 CONTINUE                                                          DA03380*
C                                                                       DA03390*
C     ---READ THE ARRAY RB---                                           DA03400*
      CALL GWDBAS (0, SETNAM(35), V, 3, 1, K0, I0, J0, 0, 0, 0.)        DA03410*
C                                                                       DA03420*
      DO 610 K=1,K0                                                     DA03430*
      DO 610 I=1,I0                                                     DA03440*
      DO 610 J=1,J0                                                     DA03450*
        NN=IDR(I,J,K)                                                   DA03460*
        IF(NN.GT.0) RB(NN)=V(I,J,K)                                     DA03470*
  610 CONTINUE                                                          DA03480*
C                                                                       DA03490*
C     ---READ THE ARRAY RH---                                           DA03500*
      CALL GWDBAS (0, SETNAM(37), V, 3, 1, K0, I0, J0, 0, 0, 0.)        DA03510*
C                                                                       DA03520*
      DO 615 K=1,K0                                                     DA03530*
      DO 615 I=1,I0                                                     DA03540*
      DO 615 J=1,J0                                                     DA03550*
        NN=IDR(I,J,K)                                                   DA03560*
        IF(NN.GT.0) RH(NN)=V(I,J,K)                                     DA03570*
  615 CONTINUE                                                          DA03580*
C                                                                       DA03590*
  700 GO TO 998                                                         DA03600*
C                                                                       DA03610*
C                                                                       DA03620*
C     ..................................................................DA03630
C     **********************                                            DA03640
      ENTRY NEWPER                                                      DA03650
C     **********************                                            DA03660
C                                                                       DA03670
C...READ GROUP IV DATA FOR A NEW STRESS PERIOD...                       DA03680*
C                                                                       DA03690*
      READ (5,331) KP, NUMT, TMAX, DELT, CDLT, IWEL, IRECH, IRIV, IHED  DA03700*
C                                                                       DA03710*
C     ---COMPUTE ACTUAL DELT AND NUMT---                                DA03720
      DT=DELT/24.                                                       DA03730
      TM=0.0                                                            DA03740
      DO 220 I=1,NUMT                                                   DA03750
      J=I                                                               DA03760*
      DT=CDLT*DT                                                        DA03770
      TM=TM+DT                                                          DA03780
      IF (TM.GE.TMAX) GO TO 230                                         DA03790
  220 CONTINUE                                                          DA03800
      GO TO 240                                                         DA03810
  230 DELT=TMAX/TM*DELT                                                 DA03820
      NUMT=J                                                            DA03830*
C                                                                       DA03840
  240 WRITE (6,400) KP,NUMT,TMAX,DELT,CDLT,IWEL,IRECH,IRIV,IHED         DA03850*
```

```
      KPM1=KPM1+1
      IF(KP.NE.KPM1) WRITE(6,9901) KPM1
      KP=KPM1
      DELT=DELT*3600./CDLT
      TMAX=TMAX*86400.
      SUMP=0.0
      NWEL=IWEL
      STIME=SUM+DELT*CDLT
      DELT1=DELT
      ISTEDY=0
C
C...READ NEW DISCHARGE OR RECHARGE RATES FOR CONSTANT-FLUX CELLS.
      IF(IWEL.LE.0) GO TO 250
      READ(5,332) SETID
      CALL GWDBAS(0, SETID, WELL, 3, 1, K0, I0, J0, 0, 1, STIME)
        DZ=1.
      DO 245 K=1,K0
      IF(IEQN.EQ.ICHK(11)) DZ=DELZ(K)
      DO 245 J=1,J0
        DX=DELX(J)*DZ
      DO 245 I=1,I0
  245 WELL(I,J,K)=WELL(I,J,K)/(DELY(I)*DX)
C
C...READ NEW RECHARGE RATES FOR THE MODEL.
  250 IF(IQRE.NE.ICHK(7) .OR. IRECH.EQ.0) GO TO 255
      READ(5,332) SETID
      CALL GWDBAS(0,SETID, QRE, 2, 1, 1, I0, J0, 0, 1, STIME)
      IF(IEQN.NE.ICHK(11)) GO TO 255
        DO 251 J=1,J0
        DO 251 I=1,I0
        K=LAYRCH(I,J)
        IF(K.GT.0) QRE(I,J)=QRE(I,J)/DELZ(K)
  251 CONTINUE
C
C...READ NEW HEADS FOR LEAKY RIVER CELLS.
  255 IF(NRIV.LE.0 .OR. IRIV.LE.0) GO TO 260
      READ(5,332) SETID
      DO 256 K=1,K0
      DO 256 I=1,I0
      DO 256 J=1,J0
        V(I,J,K)=0.
        NN=IDR(I,J,K)
        IF(NN.GT.0) V(I,J,K)=RH(NN)
  256 CONTINUE
C
      CALL GWDBAS(0, SETID, V, 3, 1, K0, I0, J0, 0, 1, STIME)
C
      DO 258 K=1,K0
      DO 258 I=1,I0
      DO 258 J=1,J0
        NN=IDR(I,J,K)
        IF(NN.GT.0) RH(NN)=V(I,J,K)
  258 CONTINUE
C
C...READ NEW HEADS FOR CONSTANT-HEAD CELLS.
```

```
  260 IF(NCH.LE.0 .OR. IHED.LE.0) GO TO 270                            DA04410*
      READ(5,332) SETID                                                 DA04420*
      DO 262 K=1,K0                                                     DA04430*
      DO 262 J=1,J0                                                     DA04440*
      DO 262 I=1,I0                                                     DA04450*
  262 V(I,J,K)=PHI(I,J,K)                                               DA04460*
C                                                                       DA04470*
      CALL GWDBAS(0, SETID, V, 3, 1, K0, I0, J0, 0, 1, STIME)           DA04480*
C                                                                       DA04490*
      DO 264 K=1,K0                                                     DA04500*
      DO 264 J=1,J0                                                     DA04510*
      DO 264 I=1,I0                                                     DA04520*
        IF(S(I,J,K).LT.0.) PHI(I,J,K)=V(I,J,K)                          DA04530*
  264 CONTINUE                                                          DA04540*
C                                                                       DA04550*
  270 CONTINUE                                                          DA04560*
C                                                                       DA04570*
  998 RETURN                                                            DA04580*
  999 STOP                                                              DA04590*
C                                                                       DA04600*
C                                                                       DA04610*
  330 FORMAT (3I5, 5F5.0, 4I5)                                          DA04620*
  331 FORMAT(2I5, 3F5.0, 4I5)                                           DA04630*
  332 FORMAT(2A4)                                                       DA04640*
  340 FORMAT(1H-, 'DATA FROM RECORD 9 IN GROUP I DATA:'/1X,35(1H-)/     DA04650*
     1 6X, 'NPER   = ',I5 / 6X, 'KTH    = ',I5 / 6X, 'LENGTH = ',       DA04660*
     2 I5 / 6X, 'ERR    = ',F14.8 / 6X, 'ERMULT = ',F14.8 /             DA04670*
     3 6X, 'WMAX1  = ',F14.8 / 6X, 'BETA1  = ',F14.8 / 6X, 'STEADY = ', DA04680*
     4 F14.8 / 6X, 'ITRHED = ',I5 / 6X, 'IPHED  = ',I5 )                DA04690*
  350 FORMAT ('0',I3,1X,20F6.1/(5X,20F6.1))                             DA04700*
  360 FORMAT (8F10.0)                                                   DA04710*
  391 FORMAT(/ '-*** COUNT OF CONSTANT-HEAD CELLS = ',I5,               DA04720*
     1  24H  (USER SPECIFICATION = ,I5, 5H) ***/)                       DA04730*
  392 FORMAT( '-***ERROR*** THE ABOVE COUNT EXCEEDS THE USER SPECIFICATIDA04740*
     1ON.' )                                                            DA04750*
  393 FORMAT(/ '-*** COUNT OF LEAKY-RIVER CELLS = ',I5,                 DA04760*
     1  24H  (USER SPECIFICATION = ,I5, 5H) ***/)                       DA04770*
  394 FORMAT(/ '-*** COUNT OF DRAIN CELLS = ',I5,                       DA04780*
     1  24H  (USER SPECIFICATION = ,I5, 5H) ***/)                       DA04790*
  400 FORMAT(1H-, 'DATA FROM RECORD 1 IN GROUP IV DATA: PARAMETERS FOR TDA04800*
     1HE START OF A NEW STRESS PERIOD' / 1X, 84(1H-) /                  DA04810*
     2 6X, 'STRESS PERIOD NO. (KP)', 33X,3H = ,I7 /                     DA04820*
     3 6X, 'NUMBER OF TIME STEPS (NUMT)', 28X,3H = ,I7 /                DA04830*
     4 6X, 'TOTAL SIMULATION TIME (TMAX)',27X,3H = ,F12.4, 5H DAYS /    DA04840*
     5 6X, 'LENGTH OF FIRST TIME STEP (DELT)',23X,3H = ,F12.4,6H HOURS /DA04850*
     6 6X, 'MULTIPLIER FOR DELT (CDLT)',29X,3H = ,F12.4 /               DA04860*
     7 6X, 'CODE FOR READING NEW VALUES OF CONSTANT FLUX (IWEL)',4X,    DA04870*
     8   3H = ,I7 /                                                     DA04880*
     9 6X, 'CODE FOR READING NEW VALUES OF RECHARGE (IRECH)',8X,3H = ,  DA04890*
     X   I7 /                                                           DA04900*
     1 6X, 'CODE FOR READING NEW VALUES OF HEAD FOR RIVERS (IRIV)',2X,  DA04910*
     2   3H = ,I7 /                                                     DA04920*
     3 6X, 'CODE FOR READING NEW VALUES OF CONSTANT HEAD (IHED)',4X,    DA04930*
     4   3H = ,I7 )                                                     DA04940*
  430 FORMAT (1H-,  'READ RECORDS 1, 2, AND 3 OF GROUP II DATA FROM FORTDA04950*
```

```
     1RAN FILE NO. ',I2 / 1X, 65(1H-) /                               DA04960*
     2 6X, 'DURATION OF PREVIOUS SIMULATION (SUM)',23X,3H = ,F14.2 /   DA04970*
     3 6X, 'DURATION OF LAST STRESS PERIOD IN PREVIOUS SIMULATION (SUMP)DA04980*
     4',3H = ,F14.2 )                                                  DA04990*
  431 FORMAT (1H-,  'READ RECORDS 4 AND 5 OF GROUP II DATA FROM FORTAN FDA05000*
     1ILE NO. ',I2 / 1X, 61(1H-) /                                     DA05010*
     2 6X, 'DURATION OF PREVIOUS SIMULATION (SUM)',23X,3H = ,F14.2 /   DA05020*
     3 6X, 'DURATION OF LAST STRESS PERIOD IN PREVIOUS SIMULATION (SUMP)DA05030*
     4',3H = ,F14.2 )                                                  DA05040*
  440 FORMAT (1H1, 'DATA FROM RECORD 3 OR 4 IN GROUP II DATA: INITIAL HEDA05050*
     1AD FOR LAYER',I3, '. PREVIOUS SIMULATION TIME = ',F14.2 /        DA05060*
     2 1X, 110(1H-) / 4HOROW )                                         DA05070*
  450 FORMAT (4F20.0)                                                  DA05080*
  460 FORMAT (5F10.0, A8)                                              DA05090*
  461 FORMAT(80I1)                                                     DA05100*
  462 FORMAT(1H-,5X, 'PRINT DRAWDOWN VALUES (OR MAPS) FOR THE FOLLOWING DA05110*
     1LAYERS:' // 7X, 40I3)                                            DA05120*
  463 FORMAT(1H-,5X, 'PRINT HEAD VALUES (OR MAPS) FOR THE FOLLOWING LAYEDA05130*
     1RS:' // 7X, 40I3)                                                DA05140*
  470 FORMAT(1H-, 'DATA FROM RECORDS 10 AND 11 IN GROUP I DATA: PARAMETEDA05150*
     1RS FOR PRINTING MAPS' / 1X, 73(1H-) /                            DA05160*
     2 6X, 'CONVERSION FACTOR FOR LENGTH UNITS IN X-DIRECTION OF SCALED DA05170*
     2MAPS (XSCALE)', 3H = ,F12.4 /                                    DA05180*
     3 6X, 'CONVERSION FACTOR FOR LENGTH UNITS IN Y-DIRECTION OF SCALED DA05190*
     3MAPS (YSCALE)', 3H = ,F12.4 /                                    DA05200*
     4 6X, 'NUMBER OF LENGTH UNITS PER INCH ON SCALED MAPS (DINCH)',   DA05210*
     4    19X,3H = , F12.4 /                                           DA05220*
     5 6X, 'MULTIPLIER FOR DRAWDOWN PLOTTED ON SCALED MAPS (FACT1)',   DA05230*
     5    19X,3H = ,F12.4 /                                            DA05240*
     6 6X, 'MULTIPLIER FOR HEAD PLOTTED ON SCALED MAPS (FACT2)',       DA05250*
     6    23X,3H = , F12.4 /                                           DA05260*
     7 6X, 'NAME OF LENGTH UNIT FOR SCALED MAPS (MESUR)',30X,3H = , A8) DA05270*
 9901 FORMAT('-***ERROR ON GROUP IV DATA RECORD:  VALUE OF KP RESET TO 'DA05280*
     1,I5, 4H *** )                                                    DA05290*
      END                                                              DA05300
```

```
      SUBROUTINE ERMES1 (IRTCOD, IERNO, A)                              ER00010*
C...PRINT AN ERROR MESSAGE IF A PROGRAM INTERRUPT OCCURS BECAUSE OF     ER00020*
C   ONE OF THE FOLLOWING FLOATING-POINT OPERATIONS:                     ER00030*
C     1. OVERFLOW (ERROR NO. 207)                                       ER00040*
C     2. UNDERFLOW (ERROR NO. 208)                                      ER00050*
C     3. DIVISION BY ZERO (ERROR NO. 209)                               ER00060*
C   THIS ROUTINE IS CALLED FROM ERROR-HANDLING ROUTINES IN THE          ER00070*
C   FORTRAN LIBRARY.                                                    ER00080*
C                                                                       ER00090*
C   NOTE:  SET IRTCOD=0 ONLY IF THE VALUE OF "A" IS NOT RESET           ER00100*
C          IN THIS ROUTINE.                                             ER00110*
C                                                                       ER00120*
      COMMON/ERDAT1/ LOC, KLOC, ILOC, JLOC                              ER00130*
C                                                                       ER00140*
      REAL*8  A, BIG, ZERO                                              ER00150*
      DATA  ZERO, BIG / 0.D0, Z7FFFFFFFFFFFFFFF /                       ER00160*
C                                                                       ER00170*
      IF(IRTCOD.LE.0 .OR. IERNO.LT.207 .OR. IERNO.GT.209) GO TO 20      ER00180*
C         IF(IERNO.EQ.207) A=BIG                                        ER00190*
C         IF(IERNO.EQ.208) A=ZERO                                       ER00200*
C         IF(IERNO.EQ.209) A=ZERO                                       ER00210*
C         WRITE(6,9001) IERNO, LOC, KLOC, ILOC, JLOC, A                 ER00220*
          WRITE(6,9000) IERNO, LOC, KLOC, ILOC, JLOC                    ER00230*
          IRTCOD=0                                                      ER00240*
          GO TO 999                                                     ER00250*
C                                                                       ER00260*
   20 WRITE(6,9002) IERNO                                               ER00270*
      IRTCOD=0                                                          ER00280*
C                                                                       ER00290*
  999 RETURN                                                            ER00300*
C                                                                       ER00310*
C                                                                       ER00320*
 9000 FORMAT( '0*** USER ACTION FOR IBM ERROR NO. ',I3,                 ER00330*
     1 '  AT PROGRAM LOCATION = ',I3,                                   ER00340*
     2 ', CELL COORDINATES (K,I,J) = (', I3,1H,, I3,1H,, I3, 2H): //    ER00350*
     3 8X,  'VALUE IN RESULT REGISTER NOT RESET BY USER - IBM STANDARD ER00360*
     4RESET IS USED.' )                                                 ER00370*
 9001 FORMAT( '0*** USER ACTION FOR IBM ERROR NO. ',I3,                 ER00380*
     1 '  AT PROGRAM LOCATION = ',I3,                                   ER00390*
     2 ', CELL COORDINATES (K,I,J) = (', I3,1H,, I3,1H,, I3, 2H): //    ER00400*
     3 8X,  'VALUE IN RESULT REGISTER RESET BY USER TO ',1P,E10.3 )     ER00410*
 9002 FORMAT( '0*** USER ACTION FOR IBM ERROR NO. ',I3,                 ER00420*
     1 ':  NO ACTION TAKEN BY USER.')                                   ER00430*
      END                                                               ER00440*
```

```
      SUBROUTINE FWSW (IOPT, PHI, T, WELL, PERM, BOTTOM, IDR, RH, RC,   FW00010*
     1   INT, TT, DBOT, BTM, ISEA, RC2, WEL2, ZOLD, S, EL, FL, GL, IDN) FW00020*
C                                                                       FW00030*
C...LOCATE THE FRESHWATER-SALTWATER INTERFACE FOR THE USGS              FW00040*
C   3-D FINITE-DIFFERENCE GROUND-WATER FLOW MODEL.                      FW00050*
C                                                                       FW00060*
C   THE INTERFACE IS DEFINED AS A SURFACE ALONG WHICH PRESSURE          FW00070*
C   IN THE FRESHWATER ZONE EQUALS PRESSURE IN THE SALTWATER             FW00080*
C   ZONE.  FLUID IN THE SALTWATER ZONE IS STATIC BECAUSE THE            FW00090*
C   MODEL DOES NOT SOLVE THE EQUATIONS OF FLUID MOTION IN THIS          FW00100*
C   ZONE.  THIS ROUTINE DETERMINES IF THE INTERFACE IS ABOVE A          FW00110*
C   MODEL CELL, WITHIN A CELL, OR BELOW A CELL.  IF THE INTERFACE       FW00120*
C   IS ABOVE A CELL, THE CELL TRANSMISSIVITY IS SET TO ZERO.  IF        FW00130*
C   THE INTERFACE IS WITHIN A CELL, A NEW CELL TRANSMISSIVITY IS        FW00140*
C   CALCULATED USING THE THICKNESS OF FRESHWATER IN THE CELL.           FW00150*
C   IF THE INTERFACE IS BELOW A CELL, THE CELL TRANSMISSIVITY           FW00160*
C   IS NOT CHANGED.                                                     FW00170*
C                                                                       FW00180*
C   THIS ROUTINE IS CALLED WHEN ISALT=1.                                FW00190*
C                                                                       FW00200*
C...WRITTEN BY:  D.B. SAPIK, USGS-WRD, TACOMA, WASHINGTON (SEPT. 1982). FW00210*
C                                                                       FW00220*
      REAL*8  PHI                                                       FW00230*
C                                                                       FW00240*
      DIMENSION  PHI(I0,J0,K0), T(I0,J0,K0), WELL(I0,J0,K0), PERM(IP,JP)FW00250*
     1, BOTTOM(IP,JP), IDR(IA1,JA1,KA1), RH(1), RC(1), INT(I0,J0,K0),   FW00260*
     2  TT(I0,J0,K0), DBOT(I0,J0,K0), BTM(IP,JP), ISEA(I0,J0,K0),       FW00270*
     3  RC2(1), WEL2(I0,J0,K0), ZOLD(I0,J0,K0), S(I0,J0,K0),            FW00280*
     4  EL(I0,J0,K0), FL(I0,J0,K0), GL(I0,J0,K0), IDN(IA2,JA2,KA2),     FW00290*
     5  SETNAM(6)                                                       FW00300*
C                                                                       FW00310*
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,       NPER,KTH,ITMAX,LENGTH,KP,NFW00320*
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCFW00330*
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN   FW00340*
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC   FW00350*
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD       FW00360*
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY                               FW00370*
C                                                                       FW00380*
      COMMON /SPARAM/ TMAX, CDLT, DELT, ERR, TEST, SUM, SUMP,           FW00390*
     1   BETA1, WMAX1, ERMULT, STEADY, DELT1                            FW00400*
C                                                                       FW00410*
      DATA INIT, IBLANK /0, 1H /                                        FW00420*
     1,     SETNAM /4HDBOT,1H , 4HINTR,4HFACE, 4HINTR,4HUCOD/           FW00430*
C                                                                       FW00440*
C...DEFINITION OF VARIABLES USED IN PROGRAM:                            FW00450*
C     ZI     = DEPTH BELOW SEA LEVEL TO INTERFACE IN THIS ITERATION.    FW00460*
C     ZOLD   = DEPTH TO THE INTERFACE IN THE PREVIOUS ITERATION.        FW00470*
C     ZTOP   = DEPTH BELOW SEA LEVEL TO TOP OF A CELL.                  FW00480*
C     ZBOT   = DEPTH BELOW SEA LEVEL TO BOTTOM OF A CELL.               FW00490*
C     INT    = INTRUSION CODES USED FOR KEEPING TRACK OF MODEL          FW00500*
C              CELLS THAT ARE TOTALLY INTRUDED BY SALTWATER, PARTIALLY  FW00510*
C              INTRUDED, OR NOT INTRUDED.                               FW00520*
C            =  0 IF T=0 AT START OF SIMULATION.                        FW00530*
C            =  1 IF T>0 AT START OF SIMULATION AND CELL IS NOT INTRUDEDFW00540*
C            = -1 IF T SET TO 0 BECAUSE OF TOTAL INTRUSION DURING       FW00550*
```

```
C                SIMULATION.                                         FW00560*
C           = -2 IF T IS ADJUSTED BECAUSE OF PARTIAL INTRUSION       FW00570*
C                DURING SIMULATION.                                  FW00580*
C           =  3 FOR A WELL CELL WITH T>0 AND NO INTRUSION.          FW00590*
C           = -3 FOR A WELL CELL PARTLY INTRUDED BY SALTWATER AND T  FW00600*
C                IS ADJUSTED.                                        FW00610*
C           =  4 FOR A LEAKY-OCEAN CELL WITH T>0 AND NO INTRUSION.   FW00620*
C           = -4 FOR A LEAKY-OCEAN CELL PARTLY INTRUDED BY SALTWATER FW00630*
C                AND T IS ADJUSTED.                                  FW00640*
C           =  5 FOR A LEAKY-RIVER CELL WITH T>0 AND NO INTRUSION.   FW00650*
C           = -5 FOR A LEAKY-RIVER CELL PARTLY INTRUDED BY SALTWATER FW00660*
C                AND T IS ADJUSTED.                                  FW00670*
C                                                                    FW00680*
C--------------------------------------------------------------------FW00690*
C                                                                    FW00700*
C...NOW WHERE DO I GO?                                               FW00710*
      IF(IOPT.EQ.1) GO TO 100                                        FW00720*
      IF(IOPT.GE.2) GO TO 200                                        FW00730*
C                                                                    FW00740*
C...MODEL MUST HAVE MORE THAN 1 LAYER.                               FW00750*
      IF(K0.LT.2) GO TO 901                                          FW00760*
      INIT=1                                                         FW00770*
C                                                                    FW00780*
C...READ RECORD 1 OF GROUP III DATA.                                 FW00790*
      READ(5,8000) RF, NTAD, ISERCH, LFRESH, ISWDIV, ITSTRT, ISOLAT,  FW00800*
     1 ZIMAX, ZDAMP, THKMIN, THKPCT, THKFUL, THKRST, IREST, MESG1,    FW00810*
     2 INT1PR, INTVAL, INTMAP, INTSAV, IPEQLB, INTCAL, INTSA2         FW00820*
          IF(RF.LE.0.) RF=40.                                        FW00830*
          IF(NTAD.LE.0) NTAD=1                                       FW00840*
          IF(ISWDIV.LE.0 .OR. ISWDIV.GT.J0) ISWDIV=J0                FW00850*
          IF(ISWDIV.EQ.J0) ISERCH=0                                  FW00860*
          IF(ITSTRT.LE.0) ITSTRT=1                                   FW00870*
          ISOLAT=MIN0(5, ISOLAT)                                     FW00880*
          ZIMAX=ABS(ZIMAX)                                           FW00890*
          ZSAVE=ZDAMP                                                FW00900*
          THKMIN=AMAX1(0., THKMIN)                                   FW00910*
          THKPCT=AMAX1(1.E-5, AMIN1(1., THKPCT))                     FW00920*
          IF(THKFUL.LE.0.) THKFUL=1.-10.*THKPCT                      FW00930*
          IF(THKRST.LE.0.) THKRST=5.*THKPCT                          FW00940*
      WRITE(6,9000) RF, NTAD, ISERCH, LFRESH, ISWDIV, ITSTRT, ISOLAT, FW00950*
     1 ZIMAX, ZDAMP, THKMIN, THKPCT, THKFUL, THKRST, IREST, MESG1,    FW00960*
     2 INT1PR, INTVAL, INTMAP, INTSAV, IPEQLB, INTCAL, INTSA2         FW00970*
C                                                                    FW00980*
      IF(LFRESH.LE.0 .OR. LFRESH.GE.K0) GO TO 902                    FW00990*
C                                                                    FW01000*
C...READ RECORD 2 OF GROUP III DATA: DEPTHS BELOW SEA LEVEL TO BOTTOM FW01010*
C  OF EACH CELL.                                                     FW01020*
      CALL GWDBAS (0, SETNAM, DBOT, 3, 1, K0, I0, J0, 0, 0, 0.)      FW01030*
C                                                                    FW01040*
C...SAVE ARRAYS THAT ARE CHANGED WHEN CELLS ARE INTRUDED.            FW01050*
      DO 6 K=1,K0                                                    FW01060*
      DO 6 J=1,J0                                                    FW01070*
      DO 6 I=1,I0                                                    FW01080*
        TT(I,J,K)=T(I,J,K)                                           FW01090*
        IF(K.NE.LWTABL) GO TO 6                                      FW01100*
```

```
         TT(I,J,K)=PERM(I,J)                                            FW01110*
         BTM(I,J)=BOTTOM(I,J)                                           FW01120*
    6  WEL2(I,J,K)=WELL(I,J,K)                                          FW01130*
C                                                                       FW01140*
      IF(NRIV.LE.0) GO TO 11                                            FW01150*
      DO 10 J=1,NRIV                                                    FW01160*
   10 RC2(J)=RC(J)                                                      FW01170*
   11 CONTINUE                                                          FW01180*
C                                                                       FW01190*
C...READ STARTING VALUES OF INTERFACE DEPTHS (ZOLD) -- SAVED AT THE     FW01200*
C   END OF A PREVIOUS SIMULATION.                                       FW01210*
      IF(INTCAL.LE.0) GO TO 17                                          FW01220*
        IDSK=18                                                         FW01230*
        REWIND IDSK                                                     FW01240*
        WRITE(6,9016) IDSK                                              FW01250*
      DO 15 K=1,K0                                                      FW01260*
   15 READ(IDSK) ((ZOLD(I,J,K), I=1,I0), J=1,J0)                        FW01270*
   17 CONTINUE                                                          FW01280*
C                                                                       FW01290*
C...COMPUTE STARTING VALUES OF INT, AND FOR INTRUDED CELLS, RECOMPUTE   FW01300*
C   VALUES FOR T, WELL, RC, AND BOTTOM.                                 FW01310*
      DO 28 K=1,K0                                                      FW01320*
        KP1=K+1                                                         FW01330*
      DO 28 J=1,J0                                                      FW01340*
      DO 28 I=1,I0                                                      FW01350*
        SPHI=PHI(I,J,K)                                                 FW01360*
        ZTOP=DBOT(I,J,KP1)                                              FW01370*
        ZBOT=DBOT(I,J,K)                                                FW01380*
      IF(K.NE.LWTABL) GO TO 21                                          FW01390*
        ZTOP=-SPHI                                                      FW01400*
        ZBOT=-BOTTOM(I,J)                                               FW01410*
   21 IF(INTCAL.LE.0) ZOLD(I,J,K)=RF*SPHI                               FW01420*
        ZI=ZOLD(I,J,K)                                                  FW01430*
        THKF=T(I,J,K)                                                   FW01440*
      IF(THKF.GT.0. .AND. ZI.GT.ZTOP) GO TO 22                          FW01450*
C     -- CELL IS TOTALLY INTRUDED OR STARTING T=0 --                    FW01460*
        N=-1                                                            FW01470*
        IF(THKF.LE.0.) N=0                                              FW01480*
        T(I,J,K)=0.                                                     FW01490*
        IF(K.EQ.LWTABL) PERM(I,J)=0.                                    FW01500*
        GO TO 28                                                        FW01510*
   22 WELX=WELL(I,J,K)                                                  FW01520*
      NR=0                                                              FW01530*
      ND=0                                                              FW01540*
      IF(NRIV.GT.0 .AND. IDR(I,J,K).GT.0) NR=IDR(I,J,K)                 FW01550*
      IF(NR.GT.0 .AND. ISEA(I,J,K).GT.0) ND=1                           FW01560*
      IF(ZI.LT.ZBOT) GO TO 23                                           FW01570*
C     -- CELL IS NOT INTRUDED --                                        FW01580*
        N=1                                                             FW01590*
        IF(WELX.LT.0.) N=3                                              FW01600*
        IF(ND.GT.0) N=4                                                 FW01610*
        IF(ND.EQ.0 .AND. NR.GT.0) N=5                                   FW01620*
        GO TO 28                                                        FW01630*
C     -- CELL IS PARTLY INTRUDED --                                     FW01640*
   23 IF(K.NE.LWTABL) GO TO 24                                          FW01650*
```

```
        ZBOT=ZI
        BOTTOM(I,J)=-ZI
   24 THK1=(ZI-ZTOP)/(ZBOT-ZTOP)
      T(I,J,K)=THKF*THK1
        N=-2
      IF(WELX.GE.0.) GOTO 25
        N=-3
        WELL(I,J,K)=WELX*THK1
   25 IF(NR.EQ.0) GO TO 28
        N=-4
        IF(ND.EQ.0) N=-5
        RC(NR)=RC(NR)*THK1
   28 INT(I,J,K)=N
C
C...FIND UPPERMOST LAYER AT THE BOTTOM OF THE MODEL TO BE USED FOR THE
C   FIRST ITERATION OF THE INTERFACE-LOCATION ROUTINE.
        LBOT=LFRESH
      DO 35 K=1,LFRESH
      DO 35 J=2,J1
      DO 35 I=2,I1
        IF(INT(I,J,K).EQ.0) GO TO 35
          LBOT=K
          GO TO 36
   35 CONTINUE
   36 LSTRT=LBOT
      WRITE(6,9007) LSTRT
C
C...PRINT THE ARRAY INT FOR EACH LAYER.
      IF(INT1PR.LE.0) GO TO 34
      DO 33 K=LBOT,LFRESH
        WRITE(6,9005) K
      DO 30 J=1,J0
      DO 30 I=1,I0
        IF(INT(I,J,K).NE.0 .AND. INT(I,J,K).NE.-1) GO TO 31
   30 CONTINUE
        WRITE(6,2930)
        GO TO 33
   31   WRITE(6,2926)
      DO 32 I=1,I0
  32  WRITE(6,9006) I, (INT(I,J,K), J=1,J0)
  33  CONTINUE
  34  CONTINUE
C
C...DETERMINE SEGMENT OF MODEL TO BE SEARCHED.
      JW=2
      JE=J1
      IF(ISERCH-1) 45, 43, 44
   43 JE=ISWDIV
      GO TO 45
   44 JW=ISWDIV
   45 ISWDIV=ISWDIV+J1
C
      GO TO 998
C
C-----------------------------------------------------------------------
```

```
C
C...LOCATE THE FRESHWATER-SALTWATER INTERFACE.
  100 IF(INIT.EQ.0) GO TO 900
      INIT=2
      WRITE(6,9002) KP, KT, NTRIAL
        NT1=0
        NT2=0
        NT3=0
        IERR=0
        MCNT=0
        IF(MESG1.LE.0) MCNT=1
        IF(NTRIAL.LE.0) MCNT=MESG1
        ZDAMP=ZSAVE
        IF(NTRIAL.LT.ITSTRT) ZDAMP=0.
        ZMAX=0.
        NZMAX=0
        IZ=0
        JZ=0
        KZ=0
        ZMAX2=0.
        NZMAX2=0
        IZ2=0
        JZ2=0
        KZ2=0
C
      DO 2300 JJ=JW,JE
        J=JJ
        IF(ISERCH.EQ.2) J=ISWDIV-J
      DO 2300 I=2,I1
        KSAV=0
        KADJ=0
        K=LFRESH+1
      DO 2280 L=LSTRT,LFRESH
        K=K-1
        INTX=INT(I,J,K)
        ZI=ZOLD(I,J,K)
        IF(INTX.EQ.0 .OR. (INTX.EQ.-1 .AND. (IREST.LE.0 .OR.
     1    K.EQ.LWTABL))) GO TO 2280
        IF(T(I,J,K).LE.0. .AND. INTX.NE.-1) GO TO 2210
          KP1=K+1
          ND=0
          INTRUS=0
          SPHI=PHI(I,J,K)
          ZTOP=DBOT(I,J,KP1)
          ZBOT=DBOT(I,J,K)
          IF(K.NE.LWTABL) GO TO 2201
            ZTOP=-SPHI
            ZBOT=-BTM(I,J)
 2201     WELX=WEL2(I,J,K)
          ZOLDX=ZI
          ZI=ZTOP
C
C...CHECK IF CURRENT CELL IS A LEAKY-OCEAN CELL.
C   NOTE THAT RH(NR) IS FRESHWATER HEAD IN SALTWATER ZONE,
C   RH(NR)=RB(NR)/RF, AND RB(NR)=ZTOP.  IF SPHI<RH(NR) THEN ZI<ZTOP
```

```
C   AND INTRUSION WILL OCCUR.                                           FW02760*
        IF(NRIV.LE.0 .OR. IDR(I,J,K).LE.0) GO TO 110                    FW02770*
          ND=2                                                          FW02780*
          IF(ISEA(I,J,K).GT.0) ND=1                                     FW02790*
          NR=IDR(I,J,K)                                                 FW02800*
 110   CONTINUE                                                         FW02810*
C                                                                       FW02820*
C...COMPUTE DEPTH TO INTERFACE.                                         FW02830*
          IF(INTX.NE.-1) CALL ZFWSW (ZI,ZOLDX,ZTOP,ZBOT,ZDAMP,RF,SPHI,  FW02840*
     1      THKMIN,THKPCT,THKRST,THKFUL)                                FW02850*
C                                                                       FW02860*
C...RESTORE A CELL THAT PREVIOUSLY CONTAINED SALTWATER.  USE HEAD VALUE FW02870*
C   FOR AN OVERLYING CELL TO DETERMINE IF SALTWATER FRONT SHOULD BE     FW02880*
C   LOCATED BELOW TOP OF CURRENT CELL.  IF CELL IN OVERLYING CONTIGUOUS FW02890*
C   LAYER WAS RESTORED THIS ITERATION, DO NOT RESTORE THE CURRENT CELL. FW02900*
      IF(INTX.NE.-1 .OR. KSAV.EQ.0 .OR. KP1.EQ.KADJ) GO TO 115          FW02910*
        SPHI=PHI(I,J,KSAV)                                              FW02920*
        ZOLDX=AMIN1(ZOLD(I,J,KSAV), ZTOP)                               FW02930*
        CALL ZFWSW(ZI,ZOLDX,ZTOP,ZBOT,ZDAMP,RF,SPHI,                    FW02940*
     1     THKMIN,THKPCT,THKRST,THKFUL)                                 FW02950*
        IF(ZI.LE.ZTOP) GO TO 115                                        FW02960*
        KADJ=K                                                          FW02970*
        SPHI=ZI/RF                                                      FW02980*
        PHI(I,J,K)=SPHI                                                 FW02990*
        ZOLD(I,J,K)=ZOLDX                                               FW03000*
        INTRUS=1                                                        FW03010*
  115 CONTINUE                                                          FW03020*
C                                                                       FW03030*
C...CHECK FOR INTERFACE ABOVE TOP OF AQUIFER CELL.                      FW03040*
        IF(ZI.LE.ZTOP) GO TO 2210                                       FW03050*
C                                                                       FW03060*
C...CHECK FOR INTERFACE BELOW BOTTOM OF AQUIFER CELL.                   FW03070*
        IF(ZI.GE.ZBOT) GO TO 2240                                       FW03080*
C                                                                       FW03090*
C...ADJUST TRANSMISSIVITY, WELL DISCHARGE, AND SPRING DISCH. COEFFICIENTFW03100*
C   FOR PARTLY INTRUDED OR PARTLY RESTORED CELLS.                       FW03110*
      THK1=(ZI-ZTOP)/(ZBOT-ZTOP)                                        FW03120*
      T(I,J,K)=TT(I,J,K)*THK1                                           FW03130*
      IF(WELX.LT.0.) WELL(I,J,K)=WELX*THK1                              FW03140*
      IF(ND.EQ.0) GO TO 2204                                            FW03150*
      RC(NR)=RC2(NR)*THK1                                               FW03160*
C-    IF(ND.EQ.2 .AND. RB(NR).LT.-ZI) RC(NR)=0.                         FW03170*
C-    T(I,J,K)=TT(I,J,K)                                                FW03180*
 2204 IF(K.NE.LWTABL) GO TO 2208                                        FW03190*
        BOTTOM(I,J)=-ZI                                                 FW03200*
        PERM(I,J)=TT(I,J,K)                                             FW03210*
C                                                                       FW03220*
C...SET VALUE OF INT TO INDICATE PARTIAL INTRUSION OR PARTIAL RESTORING FW03230*
C   OF FRESHWATER FLOW IN A CELL.                                       FW03240*
 2208   N=-2                                                            FW03250*
        IF(WELX.LT.0.) N=-3                                             FW03260*
        IF(ND.EQ.1) N=-4                                                FW03270*
        IF(ND.EQ.2) N=-5                                                FW03280*
        IF(INTRUS.EQ.0) GO TO 2209                                      FW03290*
          IF(ND.EQ.1) N=14                                              FW03300*
```

```
          NT3=NT3+1                                                     FW03310*
          IF(ND.EQ.1) WRITE(6,2927) I, J, K                             FW03320*
          IF(WELX.LT.0.) WRITE(6,2922) I, J, K, THK1                    FW03330*
 2209   INT(I,J,K)=N                                                    FW03340*
        IF(INTRUS.EQ.0) NT1=NT1+1                                       FW03350*
        IF(MCNT.GE.MESG1) GO TO 2260                                    FW03360*
        MCNT=MCNT+1                                                     FW03370*
        IF(INTRUS.EQ.0) WRITE(6,2917) I, J, K                           FW03380*
        IF(INTRUS.GT.0 .AND. ND.NE.1 .AND. WELX.GE.0.) WRITE(6,2916) I, FW03390*
     1    J, K                                                          FW03400*
        GO TO 2260                                                      FW03410*
C                                                                       FW03420*
C...SET VALUE OF INT TO INDICATE TOTAL INTRUSION INTO A CELL.           FW03430*
 2210   IF(INTX.EQ.-1) GO TO 2280                                       FW03440*
        INT(I,J,K)=11                                                   FW03450*
        NT2=NT2+1                                                       FW03460*
        ZI=RF*SPHI                                                      FW03470*
        T(I,J,K)=0.                                                     FW03480*
        IF(K.EQ.LWTABL) PERM(I,J)=0.                                    FW03490*
        IF(ND.EQ.1) WRITE(6,2928) I, J, K                               FW03500*
        IF(WELX.LT.0.) WRITE(6,2920) I, J, K                            FW03510*
        IF(MCNT.GE.MESG1) GO TO 2280                                    FW03520*
        MCNT=MCNT+1                                                     FW03530*
        IF(ND.NE.1 .AND. WELX.GE.0.) WRITE(6,2918) I,J,K                FW03540*
        GO TO 2280                                                      FW03550*
C                                                                       FW03560*
C...SET VALUE OF INT TO INDICATE THAT CELL IS FULLY RESTORED TO PERMIT  FW03570*
C   FLOW OF FRESH GROUNDWATER.                                          FW03580*
 2240   IF(INTX.GT.0) GO TO 2260                                        FW03590*
        N=1                                                             FW03600*
        IF(WELX.LT.0.) N=3                                              FW03610*
        IF(ND.EQ.1) N=4                                                 FW03620*
        IF(ND.EQ.2) N=5                                                 FW03630*
        IF(INTRUS.EQ.0) GO TO 2241                                      FW03640*
          IF(ND.EQ.1) N=24                                              FW03650*
          NT3=NT3+1                                                     FW03660*
          IF(ND.EQ.1) WRITE(6,2927) I, J, K                             FW03670*
          IF(WELX.LT.0.) WRITE(6,2921) I, J, K                          FW03680*
 2241   INT(I,J,K)=N                                                    FW03690*
        T(I,J,K)=TT(I,J,K)                                              FW03700*
        WELL(I,J,K)=WELX                                                FW03710*
        IF(ND.GT.0) RC(NR)=RC2(NR)                                      FW03720*
        IF(K.NE.LWTABL) GO TO 2250                                      FW03730*
          PERM(I,J)=TT(I,J,K)                                           FW03740*
          BOTTOM(I,J)=BTM(I,J)                                          FW03750*
 2250   IF(MCNT.GE.MESG1) GO TO 2260                                    FW03760*
        MCNT=MCNT+1                                                     FW03770*
        IF(.NOT.(INTRUS.GT.0 .AND. (ND.EQ.1 .OR. WELX.LT.0.)))          FW03780*
     1     WRITE(6,2915) I, J, K                                        FW03790*
 2260   CONTINUE                                                        FW03800*
          IF(INTRUS.EQ.0) KSAV=K                                        FW03810*
 2280     GL(I,J,K)=ZI                                                  FW03820*
 2300 CONTINUE                                                          FW03830*
C                                                                       FW03840*
C...RESTORE LEAKY-OCEAN CELLS THAT WERE TOTALLY INTRUDED IN A PREVIOUS  FW03850*
```

```
C   ITERATION BUT COULD NOT BE RESTORED IN THE PRECEDING DO-LOOP BECAUSEFW03860*
C   THE CELLS ARE LOCATED ALONG THE OCEAN BOTTOM (T=0 FOR OVERLYING     FW03870*
C   CELLS).  A LEAKY-OCEAN CELL IS RESTORED IF THE MAXIMUM HEAD FOR A   FW03880*
C   CONTIGUOUS ACTIVE CELL (IN AN UNDERLYING LAYER OR THE SAME LAYER)   FW03890*
C   MOVES THE SALTWATER INTERFACE BELOW THE TOP OF THE LEAKY-OCEAN CELL.FW03900*
C   DO NOT RESTORE A LEAKY-OCEAN CELL THAT WAS TOTALLY INTRUDED IN THE  FW03910*
C   PRECEDING DO-LOOP (INT=11) AND DO NOT USE HEAD FOR CONTIGUOUS LEAKY-FW03920*
C   OCEAN CELLS THAT WERE INTRUDED OR RESTORED (INT=11,14,24) IN THE    FW03930*
C   PRECEDING DO-LOOP.                                                  FW03940*
C                                                                       FW03950*
      IF(IREST.NE.1 .OR. NRIV.LE.0) GO TO 2311                          FW03960*
      DO 2310 J=JW,JE                                                   FW03970*
        JM1=J-1                                                         FW03980*
        JP1=J+1                                                         FW03990*
      DO 2310 I=2,I1                                                    FW04000*
        IM1=I-1                                                         FW04010*
        IP1=I+1                                                         FW04020*
        K=LFRESH+1                                                      FW04030*
      DO 2310 L=LSTRT,LFRESH                                            FW04040*
        K=K-1                                                           FW04050*
        IF(ISEA(I,J,K).EQ.0 .OR. INT(I,J,K).NE.-1) GO TO 2310           FW04060*
        KM1=K-1                                                         FW04070*
        KP1=K+1                                                         FW04080*
        IF(KM1.LE.0) KM1=1                                              FW04090*
        SPHI=-9999.                                                     FW04100*
        IF(K.EQ.1 .OR. T(I,J,KM1).LE.0.) GO TO 2303                     FW04110*
          SPHI=PHI(I,J,KM1)                                             FW04120*
          ZOLDX=ZOLD(I,J,KM1)                                           FW04130*
 2303     IF(T(IM1,J,K).LE.0. .OR. INT(IM1,J,K).GE.10 .OR.              FW04140*
     1       T(IM1,J,KM1).LE.0. .OR. INT(IM1,J,KM1).GE.10) GO TO 2304   FW04150*
            ZTOP=PHI(IM1,J,K)                                           FW04160*
            IF(ZTOP.LE.SPHI) GO TO 2304                                 FW04170*
            SPHI=ZTOP                                                   FW04180*
            ZOLDX=ZOLD(IM1,J,K)                                         FW04190*
 2304     IF(T(IP1,J,K).LE.0. .OR. INT(IP1,J,K).GE.10 .OR.              FW04200*
     1       T(IP1,J,KM1).LE.0. .OR. INT(IP1,J,KM1).GE.10) GO TO 2305   FW04210*
            ZTOP=PHI(IP1,J,K)                                           FW04220*
            IF(ZTOP.LE.SPHI) GO TO 2305                                 FW04230*
            SPHI=ZTOP                                                   FW04240*
            ZOLDX=ZOLD(IP1,J,K)                                         FW04250*
 2305     IF(T(I,JM1,K).LE.0. .OR. INT(I,JM1,K).GE.10 .OR.              FW04260*
     1       T(I,JM1,KM1).LE.0. .OR. INT(I,JM1,KM1).GE.10) GO TO 2306   FW04270*
            ZTOP=PHI(I,JM1,K)                                           FW04280*
            IF(ZTOP.LE.SPHI) GO TO 2306                                 FW04290*
            SPHI=ZTOP                                                   FW04300*
            ZOLDX=ZOLD(I,JM1,K)                                         FW04310*
 2306     IF(T(I,JP1,K).LE.0. .OR. INT(I,JP1,K).GE.10 .OR.              FW04320*
     1       T(I,JP1,KM1).LE.0. .OR. INT(I,JP1,KM1).GE.10) GO TO 2307   FW04330*
            ZTOP=PHI(I,JP1,K)                                           FW04340*
            IF(ZTOP.LE.SPHI) GO TO 2307                                 FW04350*
            SPHI=ZTOP                                                   FW04360*
            ZOLDX=ZOLD(I,JP1,K)                                         FW04370*
 2307     IF(SPHI.EQ.-9999.) GO TO 2310                                 FW04380*
          ZTOP=DBOT(I,J,KP1)                                            FW04390*
          ZBOT=DBOT(I,J,K)                                              FW04400*
```

```
      ZOLDX=AMIN1(ZOLDX, ZTOP)                                          FW04410*
      CALL ZFWSW(ZI,ZOLDX,ZTOP,ZBOT,ZDAMP,RF,SPHI,                      FW04420*
     1  THKMIN,THKPCT,THKRST,THKFUL)                                    FW04430*
      IF(ZI.LE.ZTOP) GO TO 2310                                         FW04440*
      N=14                                                              FW04450*
      IF(ZI.GE.ZBOT) N=24                                               FW04460*
      INT(I,J,K)=N                                                      FW04470*
      NT3=NT3+1                                                         FW04480*
      SPHI=ZI/RF                                                        FW04490*
      THK1=(ZI-ZTOP)/(ZBOT-ZTOP)                                        FW04500*
      THK1=AMIN1(1., THK1)                                              FW04510*
      T(I,J,K)=TT(I,J,K)*THK1                                           FW04520*
      NR=IDR(I,J,K)                                                     FW04530*
      RC(NR)=RC2(NR)*THK1                                               FW04540*
      PHI(I,J,K)=SPHI                                                   FW04550*
      ZOLD(I,J,K)=ZOLDX                                                 FW04560*
      GL(I,J,K)=ZI                                                      FW04570*
      WRITE(6,2927) I, J, K                                             FW04580*
      IF(WEL2(I,J,K).GE.0.) GO TO 2308                                  FW04590*
        WELL(I,J,K)=WEL2(I,J,K)*THK1                                    FW04600*
        WRITE(6,2922) I,J,K, THK1                                       FW04610*
 2308 IF(K.NE.LWTABL) GO TO 2310                                        FW04620*
      PERM(I,J)=TT(I,J,K)                                               FW04630*
      BOTTOM(I,J)=-ZI                                                   FW04640*
 2310 CONTINUE                                                          FW04650*
 2311 CONTINUE                                                          FW04660*
C                                                                       FW04670*
C...CHECK FOR ISOLATED CELLS CREATED BY INTRUSION IN SURROUNDING CELLS. FW04680*
    NT4=0                                                               FW04690*
    IT=0                                                                FW04700*
    IF(ISOLAT.LT.0) GO TO 2350                                          FW04710*
 2330 JJ=0                                                              FW04720*
   DO 2340 K=LSTRT,LFRESH                                               FW04730*
    KM1=K-1                                                             FW04740*
    KP1=K+1                                                             FW04750*
   DO 2340 J=2,J1                                                       FW04760*
    JM1=J-1                                                             FW04770*
    JP1=J+1                                                             FW04780*
   DO 2340 I=2,I1                                                       FW04790*
    IF(T(I,J,K).LE.0. .OR. S(I,J,K).LT.0.) GO TO 2340                   FW04800*
    IF(NDRAIN.GT.0 .AND. IDN(I,J,K).GT.0) GO TO 2340                    FW04810*
    IF(NRIV.GT.0 .AND. IDR(I,J,K).GT.0) GO TO 2340                      FW04820*
      N=0                                                               FW04830*
      IF(K.NE.1  .AND. T(I,J,KM1).GT.0.) N=1                            FW04840*
      IF(K.NE.K0 .AND. T(I,J,KP1).GT.0.) N=N+1                          FW04850*
      IF(T(I,JM1,K).GT.0.) N=N+1                                        FW04860*
      IF(T(I,JP1,K).GT.0.) N=N+1                                        FW04870*
      IF(T(I-1,J,K).GT.0.) N=N+1                                        FW04880*
      IF(T(I+1,J,K).GT.0.) N=N+1                                        FW04890*
      IF(N.GT.ISOLAT) GO TO 2340                                        FW04900*
        INT(I,J,K)=-1                                                   FW04910*
        NT2=NT2+1                                                       FW04920*
        T(I,J,K)=0.                                                     FW04930*
        IF(K.EQ.LWTABL) PERM(I,J)=0.                                    FW04940*
        WRITE(6,2923) I, J, K, N                                        FW04950*
```

```
          IF(WEL2(I,J,K).LT.0.) WRITE(6,2920) I, J, K                   FW04960*
          JJ=JJ+1                                                       FW04970*
 2340 CONTINUE                                                          FW04980*
C                                                                       FW04990*
      IT=IT+1                                                           FW05000*
      NT4=NT4+JJ                                                        FW05010*
      IF(JJ.GT.0 .AND. IT.LT.7) GO TO 2330                              FW05020*
C                                                                       FW05030*
C...SAVE INTERFACE POSITION FOR NEXT ITERATION, CHANGE INTRUSION CODES  FW05040*
C   FOR LEAKY-OCEAN CELLS,  RESTORE WELL CELLS THAT ARE INTRUDED AND NOTFW05050*
C   SURROUNDED BY INTRUDED CELLS, AND COMPUTE THE MAX. DISTANCE BETWEEN FW05060*
C   THE DAMPED INTERFACE POSITION AND THE EQUILIBRIUM POSITION.         FW05070*
      DO 2320 K=LSTRT,LFRESH                                            FW05080*
        KM1=K-1                                                         FW05090*
        KP1=K+1                                                         FW05100*
      DO 2320 J=JW,JE                                                   FW05110*
        JM1=J-1                                                         FW05120*
        JP1=J+1                                                         FW05130*
      DO 2320 I=2,I1                                                    FW05140*
         ZI=GL(I,J,K)                                                   FW05150*
         ZOLD(I,J,K)=ZI                                                 FW05160*
         INTX=INT(I,J,K)                                                FW05170*
C...CHANGE INTRUSION CODE.                                              FW05180*
         IF(INTX.LT.10) GO TO 2313                                      FW05190*
           IF(INTX.LT.20) INTX=10-INTX                                  FW05200*
           IF(INTX.GE.20) INTX=INTX-20                                  FW05210*
 2313    IF(INTX.EQ.-1) ZOLD(I,J,K)=DBOT(I,J,KP1)                       FW05220*
C...CHECK INTRUDED WELL CELLS.                                          FW05230*
         IF(INTX.GE.0 .OR. INTX.EQ.-4 .OR. WEL2(I,J,K).GE.0.) GO TO 2315FW05240*
         IF((K.GT.1 .AND. INT(I,J,KM1).LT.0) .OR. INT(I-1,J,K).LT.0 .OR.FW05250*
     1     INT(I+1,J,K).LT.0 .OR. INT(I,JM1,K).LT.0 .OR. INT(I,JP1,K).  FW05260*
     2     LT.0) GO TO 2315                                             FW05270*
          ZOLD(I,J,K)=DBOT(I,J,K)                                       FW05280*
          WELL(I,J,K)=WEL2(I,J,K)                                       FW05290*
          T(I,J,K)=TT(I,J,K)                                            FW05300*
          IF(K.NE.LWTABL) GO TO 2314                                    FW05310*
            PERM(I,J)=TT(I,J,K)                                         FW05320*
            BOTTOM(I,J)=BTM(I,J)                                        FW05330*
 2314     INTX=3                                                        FW05340*
          IF(NRIV.LE.0 .OR. IDR(I,J,K).EQ.0) GO TO 2315                 FW05350*
            NR=IDR(I,J,K)                                               FW05360*
            RC(NR)=RC2(NR)                                              FW05370*
            INTX=5                                                      FW05380*
            IF(ISEA(I,J,K).GT.0) INTX=4                                 FW05390*
 2315     INT(I,J,K)=INTX                                               FW05400*
C...FIND MAXIMUM DISTANCE BETWEEN THE DAMPED INTERFACE POSITION AND THE FW05410*
C   EQUILIBRIUM POSITION.                                               FW05420*
      IF(INTX.EQ.-1 .OR. INTX.GT.0) GO TO 2318                          FW05430*
        THK1=RF*SNGL(PHI(I,J,K))-ZI                                     FW05440*
        IF(THK1.GE.0.) GO TO 2317                                       FW05450*
C       ---THK1 IS NEGATIVE---                                          FW05460*
        IF(ABS(THK1).GT.ZIMAX) NZMAX=NZMAX+1                            FW05470*
        IF(THK1.GE.ZMAX) GO TO 2318                                     FW05480*
          ZMAX=THK1                                                     FW05490*
          KZ=K                                                          FW05500*
```

```
          IZ=I                                                          FW05510*
          JZ=J                                                          FW05520*
          GO TO 2318                                                    FW05530*
C                                                                       FW05540*
C       ---THK1 IS POSITIVE---                                          FW05550*
 2317   IF(THK1.GT.ZIMAX) NZMAX2=NZMAX2+1                               FW05560*
        IF(THK1.LE.ZMAX2) GO TO 2318                                    FW05570*
          ZMAX2=THK1                                                    FW05580*
          KZ2=K                                                         FW05590*
          IZ2=I                                                         FW05600*
          JZ2=J                                                         FW05610*
 2318 CONTINUE                                                          FW05620*
 2320 CONTINUE                                                          FW05630*
C                                                                       FW05640*
C...PRINT SUMMARY OF INTRUDED AND RESTORED CELLS.                       FW05650*
 2350 WRITE(6,9003) NT1, NT2, NT3, IT, NT4                              FW05660*
      IF(ABS(ZMAX).GT.ZIMAX .OR. ZMAX2.GT.ZIMAX .OR. NT2.GT.0 .OR.      FW05670*
     1  NT3.GT.0) IERR=1                                                FW05680*
      IF(NTRIAL.GT.0) WRITE(6,9004) ZMAX, IZ, JZ, KZ, NZMAX, ZMAX2,     FW05690*
     1     IZ2, JZ2, KZ2, NZMAX2                                        FW05700*
C                                                                       FW05710*
      GO TO 998                                                         FW05720*
C                                                                       FW05730*
C-----------------------------------------------------------------------FW05740*
C                                                                       FW05750*
C...PRINT THE INT ARRAY AFTER CONVERGENCE OF INTERFACE-LOCATION         FW05760*
C   PROCEDURE (SUBR. FWSW).                                             FW05770*
  200 IF(INIT.NE.2) GO TO 900                                           FW05780*
C                                                                       FW05790*
      IF(INTVAL.LE.0) GO TO 261                                         FW05800*
        WRITE(6,9001)                                                   FW05810*
      DO 240 K=LBOT,LFRESH                                              FW05820*
        NT1=0                                                           FW05830*
        NT2=0                                                           FW05840*
        NT3=0                                                           FW05850*
        JJ=0                                                            FW05860*
      DO 230 I=2,I1                                                     FW05870*
      DO 230 J=2,J1                                                     FW05880*
        JJ=JJ+1                                                         FW05890*
        IF(INT(I,J,K).LT.-1) NT1=NT1+1                                  FW05900*
        IF(INT(I,J,K).EQ.-1) NT2=NT2+1                                  FW05910*
        IF(INT(I,J,K).EQ.0) NT3=NT3+1                                   FW05920*
  230 CONTINUE                                                          FW05930*
      WRITE(6,2925) K, NT2, NT1                                         FW05940*
      IF(JJ.EQ.(NT2+NT3)) GO TO 239                                     FW05950*
      WRITE(6,2926)                                                     FW05960*
      DO 238 I=1,I0                                                     FW05970*
  238 WRITE(6,9006) I, (INT(I,J,K), J=1,J0)                             FW05980*
      GO TO 240                                                         FW05990*
  239 WRITE(6,2930)                                                     FW06000*
  240 CONTINUE                                                          FW06010*
C                                                                       FW06020*
C...PRINT SUMMARY OF CELLS WITH ADJUSTED TRANSMISSIVITIES.              FW06030*
      WRITE(6,2940)                                                     FW06040*
      DO 260 K=LBOT,LFRESH                                              FW06050*
```

```
        KP1=K+1                                                        FW06060*
      DO 260 I=2,I1                                                    FW06070*
      DO 260 J=JW,JE                                                   FW06080*
        IF(INT(I,J,K).GE.-1) GO TO 260                                 FW06090*
        SPHI=PHI(I,J,K)                                                FW06100*
        ZI=ZOLD(I,J,K)                                                 FW06110*
        THK1=DBOT(I,J,K)-DBOT(I,J,KP1)                                 FW06120*
        THKF=ZI-DBOT(I,J,KP1)                                          FW06130*
        IF(K.NE.LWTABL) GO TO 259                                      FW06140*
          THK1=SPHI-BTM(I,J)                                           FW06150*
          THKF=SPHI-BOTTOM(I,J)                                        FW06160*
  259   WRITE(6,2950) K, I, J, ZI, THK1, THKF                          FW06170*
  260 CONTINUE                                                         FW06180*
  261 CONTINUE                                                         FW06190*
C                                                                      FW06200*
C...PRINT MAP OF GREATEST DEPTHS TO INTERFACE AND SAVE DEPTHS ON       FW06210*
C     DISK FILE.                                                       FW06220*
      IF(INTMAP.LE.0 .AND. INTSAV.LE.0) GO TO 256                      FW06230*
C     ---DETERMINE DEPTHS TO THE INTERFACE AND INTRUSION CODES---      FW06240*
      DO 248 J=1,J0                                                    FW06250*
      DO 248 I=1,I0                                                    FW06260*
        N=0                                                            FW06270*
        X=-99.                                                         FW06280*
      DO 246 K=LBOT,LFRESH                                             FW06290*
        INTX=INT(I,J,K)                                                FW06300*
       IF(INTX.EQ.0) GO TO 246                                         FW06310*
        X=ZOLD(I,J,K)                                                  FW06320*
       IF(INTX.NE.-1) GO TO 247                                        FW06330*
        N=1                                                            FW06340*
  246 CONTINUE                                                         FW06350*
        IF(N.GT.0) X=-77.                                              FW06360*
  247   FL(I,J,1)=X                                                    FW06370*
        FL(I,J,2)=INTX                                                 FW06380*
  248 CONTINUE                                                         FW06390*
C                                                                      FW06400*
C     ---PRINT MAP OF GREATEST DEPTHS TO INTERFACE---                  FW06410*
      IF(INTMAP.LE.0) GO TO 255                                        FW06420*
        WRITE(6,9012)                                                  FW06430*
      DO 252 I=1,I0                                                    FW06440*
  252 WRITE(6,9011) I, (FL(I,J,1), J=1,J0)                             FW06450*
C                                                                      FW06460*
C     ---PRINT MAP OF INTRUSION CODES---                               FW06470*
        WRITE(6,9014)                                                  FW06480*
      DO 254 I=1,I0                                                    FW06490*
  254 WRITE(6,9011) I, (FL(I,J,2), J=1,J0)                             FW06500*
  255 CONTINUE                                                         FW06510*
C                                                                      FW06520*
C     ---SAVE DEPTHS TO INTERFACE AND INTRUSION CODES ON DISK FILE---  FW06530*
      IF(INTSAV.LE.0) GO TO 256                                        FW06540*
        IDSK=17                                                        FW06550*
        K=1                                                            FW06560*
      WRITE(6,9013) IDSK                                               FW06570*
      WRITE(IDSK) SETNAM(3), SETNAM(4), SUM, K, I0, J0,                FW06580*
     1   ((FL(I,J,1), I=1,I0), J=1,J0)                                 FW06590*
      WRITE(IDSK) SETNAM(5), SETNAM(6), SUM, K, I0, J0,                FW06600*
```

```
     1   ((FL(I,J,2), I=1,I0), J=1,J0)                                 FW06610*
  256 CONTINUE                                                         FW06620*
C                                                                      FW06630*
C...PRINT DIFFERENCE BETWEEN EQUILIBRIUM AND COMPUTED INTERFACE POSITIONFW06640*
      IF(IPEQLB.LE.0) GO TO 266                                        FW06650*
        WRITE(6,9001)                                                  FW06660*
      DO 265 K=LBOT,LFRESH                                             FW06670*
        WRITE(6,9010) K                                                FW06680*
      DO 265 I=1,I0                                                    FW06690*
      DO 262 J=1,J0                                                    FW06700*
        FL(I,J,K)=0.                                                   FW06710*
        IF(J.LT.JW .OR. J.GT.JE) GO TO 262                             FW06720*
        IF(INT(I,J,K).NE.0 .AND. INT(I,J,K).NE.-1) FL(I,J,K)=          FW06730*
     1     RF*SNGL(PHI(I,J,K)) - ZOLD(I,J,K)                           FW06740*
  262 CONTINUE                                                         FW06750*
        WRITE(6,9011) I, (FL(I,J,K), J=1,J0)                           FW06760*
  265 CONTINUE                                                         FW06770*
  266 CONTINUE                                                         FW06780*
C                                                                      FW06790*
C...WRITE ON DISK FILE THE FINAL VALUES OF INTERFACE DEPTHS (ZOLD)     FW06800*
C   AND INTRUSION CODES (INT).                                         FW06810*
      IF(INTSA2.LE.0) GO TO 281                                        FW06820*
        IDSK=18                                                        FW06830*
        REWIND IDSK                                                    FW06840*
        WRITE(6,9015) IDSK                                             FW06850*
      DO 270 K=1,K0                                                    FW06860*
  270 WRITE(IDSK) ((ZOLD(I,J,K), I=1,I0), J=1,J0)                      FW06870*
      DO 275 K=1,K0                                                    FW06880*
  275 WRITE(IDSK) ((INT(I,J,K), I=1,I0), J=1,J0)                       FW06890*
  281 CONTINUE                                                         FW06900*
C                                                                      FW06910*
C...SAVE FINAL VALUES OF THE ARRAY INT FOR PRINTOUT IN SUBR. VSEC.     FW06920*
      DO 290 K=1,K0                                                    FW06930*
      DO 290 J=1,J0                                                    FW06940*
      DO 290 I=1,I0                                                    FW06950*
  290 FL(I,J,K)=INT(I,J,K)                                             FW06960*
C                                                                      FW06970*
      GO TO 998                                                        FW06980*
C                                                                      FW06990*
C----------------------------------------------------------------------FW07000*
C                                                                      FW07010*
C...ERROR MESSAGES THAT TERMINATE THIS JOB.                            FW07020*
  900 WRITE(6,9900)                                                    FW07030*
      GO TO 999                                                        FW07040*
  901 WRITE(6,9901)                                                    FW07050*
      GO TO 999                                                        FW07060*
  902 WRITE(6,9902)                                                    FW07070*
      GO TO 999                                                        FW07080*
C                                                                      FW07090*
C...END OF JOB.                                                        FW07100*
  998 RETURN                                                           FW07110*
  999 STOP                                                             FW07120*
C                                                                      FW07130*
C                                                                      FW07140*
 2915 FORMAT(1H0,5X, 'FRESHWATER HAS DISPLACED SALTWATER IN CELL (',   FW07150*
```

```
     1 I3,1H,, I3,1H,, I3, '). TRANSMISSIVITY RESET TO STARTING VALUE.'FW07160*
     2 )                                                               FW07170*
2916 FORMAT(1H0,5X, 'FRESHWATER HAS DISPLACED SALTWATER IN CELL (',    FW07180*
     1 I3,1H,, I3,1H,, I3, '). TRANSMISSIVITY ADJUSTED FOR PARTIAL INTRFW07190*
     2USION.' )                                                        FW07200*
2917 FORMAT(1H0,5X, 'SALTWATER INTRUSION INTO PART OF CELL (',I3,      FW07210*
     1 1H,, I3, 1H,, I3, '). TRANSMISSIVITY ADJUSTED FOR PARTIAL INTRUSFW07220*
     2ION.' )                                                          FW07230*
2918 FORMAT(1H0,5X, 'SALTWATER INTRUSION INTO ALL OF CELL (',I3,       FW07240*
     1 1H,, I3, 1H,, I3, '). TRANSMISSIVITY SET TO ZERO.')             FW07250*
2919 FORMAT(1H0,8X, 'WELL DISCHARGE FOR CELL (',I3,1H,, I3,1H,, I3,    FW07260*
     1 ') IS UNCHANGED.')                                              FW07270*
2920 FORMAT(1H0,8X, 'WELL DISCHARGE FOR CELL (',I3,1H,, I3,1H,, I3,    FW07280*
     1 ') IS SET TO ZERO.')                                            FW07290*
2921 FORMAT(1H0,8X, 'WELL DISCHARGE FOR CELL (',I3,1H,, I3,1H,, I3,    FW07300*
     1 ') IS SET TO STARTING VALUE.')                                  FW07310*
2922 FORMAT(1H0,8X, 'WELL DISCHARGE FOR CELL (',I3,1H,, I3,1H,, I3,    FW07320*
     1 ') IS SET TO STARTING VALUE MULTIPLIED BY ',1P,E10.3)           FW07330*
2923 FORMAT(1H0,8X, 'FRESHWATER CELL (', I3,1H,, I3,1H,, I3,           FW07340*
     1 ') IS SURROUNDED BY ',I1,    ' ACTIVE CELLS. TRANSMISSIVITY IS FW07350*
     2SET TO ZERO.')                                                   FW07360*
2925 FORMAT(1H-/ '-*** INTRUSION CODES AT END OF TIME STEP FOR ',      FW07370*
     1 'LAYER ',I3, 4H ***/                                            FW07380*
     2 1H0,5X, 'NO. CELLS WITH TRANSMISSIVITY SET TO ZERO SINCE START OFFW07390*
     3 SIMULATION', 3H = ,I6 / 6X, 'NO. CELLS WITH TRANSMISSIVITY ADJUSFW07400*
     3TED SINCE START OF SIMULATION',3X, 3H = ,I6)                     FW07410*
2926 FORMAT(                                                           FW07420*
     4 1H0,5X, ' 0 = TRANSMISSIVITY = 0 AT START OF SIMULATION.'/      FW07430*
     5 6X,     ' 1 = TRANSMISSIVITY > 0 AT START OF SIMULATION AND NO INFW07440*
     5TRUSION.' /                                                      FW07450*
     6 6X,     '-1 = TRANSMISSIVITY SET TO ZERO BECAUSE ALL OF CELL IS',FW07460*
     7            ' INTRUDED OR BECAUSE CELL IS ISOLATED.'/            FW07470*
     8 6X,    '-2 = TRANSMISSIVITY ADJUSTED BECAUSE PART OF CELL IS',  FW07480*
     9            ' INTRUDED.'/                                        FW07490*
     X 6X,' 3 = WELL LOCATED IN CELL WITH STARTING TRANSMISSIVITY > 0 ANFW07500*
     XD NO INTRUSION.' /                                               FW07510*
     1 6X, '-3 = WELL CELL IS PARTLY INTRUDED AND TRANSMISSIVITY IS ADJUFW07520*
     1STED.'/                                                          FW07530*
     2 6X, ' 4 = LEAKY-OCEAN CELL WITH STARTING TRANSMISSIVITY > 0 AND NFW07540*
     2O INTRUSION.' /                                                  FW07550*
     3 6X, '-4 = LEAKY-OCEAN CELL IS PARTLY INTRUDED AND TRANSMISSIVITY FW07560*
     3IS ADJUSTED.'/                                                   FW07570*
     4 6X, ' 5 = LEAKY-RIVER CELL WITH STARTING TRANSMISSIVITY > 0 AND NFW07580*
     4O INTRUSION.' /                                                  FW07590*
     5 6X, '-5 = LEAKY-RIVER CELL IS PARTLY INTRUDED AND TRANSMISSIVITY FW07600*
     5IS ADJUSTED.'/ 4H0ROW)                                           FW07610*
2927 FORMAT(1H0,8X, 'LEAKY-OCEAN CELL (',I3,1H,, I3,1H,, I3,           FW07620*
     1 ') IS RESTORED TO PERMIT FRESHWATER MOVEMENT.')                 FW07630*
2928 FORMAT(1H0,8X, 'LEAKY-OCEAN CELL (',I3,1H,, I3,1H,, I3,           FW07640*
     1 ') IS TOTALLY INTRUDED.')                                       FW07650*
2929 FORMAT(1H0,8X, 'LEAKY-OCEAN CELL (',I3,1H,, I3,1H,, I3,           FW07660*
     1 ') IS PARTLY INTRUDED.')                                        FW07670*
2930 FORMAT( '-*** ALL CELLS IN THIS LAYER ARE INTRUDED OR STARTING TRAFW07680*
     1NSMISSIVITY IS ZERO ***')                                        FW07690*
2940 FORMAT(1H1, '*** SUMMARY OF ALL CELLS WITH ADJUSTED TRANSMISSIVITFW07700*
```

```
     1IES AT END OF TIME STEP ***'/                                    FW07710*
     2 1H0, 21X, 'DEPTH TO   ORIGINAL     ADJUSTED'/                   FW07720*
     3 4X, 'LAYER  ROW  COL.  SALTWATER  THICKNESS    THICKNESS')      FW07730*
2950 FORMAT(4X,I4, 3X,I3, 2X,I3, 2X,F9.2, 2X,F9.2, 4X,F9.2)           FW07740*
8000 FORMAT(F10.0, 6I5, 6F5.0, 2I5 / 16I5)                             FW07750*
9000 FORMAT( '-DATA FROM RECORD 1 IN GROUP III DATA: PARAMETERS FOR THEFW07760*
     1 SEAWATER INTRUSION PROBLEM' /1X, 83(1H-) /                      FW07770*
     1 6X, 'RF     = ',F16.5 / 6X, 'NTAD    = ',I10/ 6X,'ISERCH = ',I10/ FW07780*
     2 6X, 'LFRESH = ',I10 / 6X, 'ISWDIV = ',I10/ 6X, 'ITSTRT = ',I10 / FW07790*
     3 6X, 'ISOLAT = ',I10 / 6X,'ZIMAX  = ',F16.5/6X,'ZDAMP  = ',F16.5/ FW07800*
     4 6X, 'THKMIN = ',F16.5/6X,'THKPCT = ',F16.5/6X,'THKFUL = ',F16.5/ FW07810*
     5 6X, 'THKRST = ',F16.5 / 6X,'IREST  = ',I10/ 6X,'MESG1  = ',I10/  FW07820*
     6 6X, 'INT1PR = ',I10/ 6X,'INTVAL = ',I10/ 6X,'INTMAP = ',I10/     FW07830*
     7 6X, 'INTSAV = ',I10/ 6X,'IPEQLB = ',I10/ 6X,'INTCAL = ',I10/     FW07840*
     8 6X, 'INTSA2 = ',I10 )                                           FW07850*
9001 FORMAT(1H1)                                                       FW07860*
9002 FORMAT(1H-/ '-*** SEARCH FOR CELLS INTRUDED BY SALTWATER DURING', FW07870*
     1  ' STRESS PERIOD NO. ',I4,    ', TIME STEP NO. ',I4,            FW07880*
     2  ', TRIAL SOLUTION NO. ',I4, 4H ***/                            FW07890*
     3 1H0,5X, 'NOTE THAT COORDINATES FOR MODEL CELLS ARE PRINTED IN THEFW07900*
     4 FORMAT (I,J,K) UNLESS OTHERWISE SPECIFIED.')                    FW07910*
9003 FORMAT(1H0,5X,  'NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAFW07920*
     1S ADJUSTED',21X, 3H = ,I5/                                       FW07930*
     2 6X,  'NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS SET TO ZFW07940*
     3ERO', 18X, 3H = ,I5/                                             FW07950*
     4 6X,  'NUMBER OF MODEL CELLS FOR WHICH TRANSMISSIVITY WAS CHANGED FW07960*
     5FROM ZERO TO NONZERO',1X, 3H = ,I5 /                             FW07970*
     6 6X,  'NUMBER OF ISOLATED CELLS FOR WHICH TRANSMISSIVITY WAS SET TFW07980*
     6O ZERO (',I3,  ' PASSES) ', 3H = ,I5 )                           FW07990*
9004 FORMAT(1H-,5X,  'MAXIMUM UPWARD MOVEMENT OF INTERFACE FOR CONVERGEFW08000*
     1NCE',4X, 3H = , 1P,E12.3, '  (I,J,K = ', I3,1H,, I3,1H,, I3,  1H)/FW08010*
     2 6X,  'NO. CELLS FOR WHICH MAX. UPWARD MOVEMENT EXCEEDS ZIMAX',2X,FW08020*
     3 3H = ,I4 /                                                      FW08030*
     4 1H0,5X, 'MAXIMUM DOWNWARD MOVEMENT OF INTERFACE FOR CONVERGENCE',FW08040*
     5 2X,3H = , 1P,E12.3,  '  (I,J,K = ', I3,1H,, I3,1H,, I3, 1H) /   FW08050*
     6 6X,  'NO. CELLS FOR WHICH MAX. DOWNWARD MOVEMENT EXCEEDS ZIMAX', FW08060*
     7 3H = ,I4 )                                                      FW08070*
9005 FORMAT(1H-/'-*** STARTING VALUES OF INTRUSION CODES FOR',         FW08080*
     1  ' LAYER ',I3, 4H *** )                                         FW08090*
9006 FORMAT(1H0,I3, 2X, 40I3 / (6X, 40I3))                             FW08100*
9007 FORMAT(  '-*** START SEARCH FOR INTERFACE IN LAYER ',I2, 4H *** ) FW08110*
9010 FORMAT(1H-/  '-*** DIFFERENCE BETWEEN EQUILIBRIUM AND COMPUTED INTFW08120*
     1ERFACE POSITIONS FOR LAYER ',I2, 4H *** / 4H0ROW )               FW08130*
9011 FORMAT(1H0, I3, 20F6.0 / (4X, 20F6.0) )                           FW08140*
9012 FORMAT( '1*** GREATEST DEPTHS TO FRESHWATER-SALTWATER INTERFACE **FW08150*
     1*' //                                                            FW08160*
     1 7X,  'VALUES > 0 ARE DEPTHS TO INTERFACE'/                      FW08170*
     2 7X,  '-77 = TOTAL INTRUSION IN A VERTICAL COLUMN OF CELLS'/     FW08180*
     3 7X,  '-99 = NO-FLOW BOUNDARY' / 4H0ROW )                        FW08190*
9013 FORMAT(  '-*** PLOT FILES FOR GREATEST INTERFACE DEPTHS AND INTRUSFW08200*
     1ION CODES WRITTEN ON DISK FILE NO. ', I2, 4H *** )               FW08210*
9014 FORMAT(  '1*** INTRUSION CODES FOR CELLS USED TO DETERMINE GREATESFW08220*
     1T DEPTHS TO FRESHWATER-SALTWATER INTERFACE ***' /                FW08230*
     2 4H0ROW )                                                        FW08240*
9015 FORMAT(  '-*** INTERFACE DEPTHS AND INTRUSION CODES AT END OF SIMUFW08250*
```

```
     1LATION ARE WRITTEN ON DISK FILE NO. ',I2, 4H *** )              FW08260*
 9016 FORMAT(  '-*** INTERFACE DEPTHS AT START OF SIMULATION ARE READ FRFW08270*
     1OM DISK FILE NO. ',I2, 4H *** )                                  FW08280*
 9900 FORMAT( '-***ERROR IN SUBR. FWSW*** INCORRECT VALUE FOR IOPT IN', FW08290*
     1 ' SUBROUTINE PARAMETER LIST.')                                  FW08300*
 9901 FORMAT( '-***ERROR IN SUBR. FWSW*** MODEL MUST HAVE AT LEAST 2 LAYFW08310*
     1ERS.')                                                           FW08320*
 9902 FORMAT( '-***ERROR IN SUBR. FWSW*** INCORRECT SPECIFICATION FOR', FW08330*
     1  ' VALUE OF LFRESH.')                                           FW08340*
      END                                                              FW08350*
```

```
      SUBROUTINE GWDBAS (IOPTN, SIDENT, A, IATYPE, NK1, NK2, NI, NJ,    GD00010*
     1                  IPRNTT, IUPDAT, STIME)                          GD00020*
C                                                                       GD00030*
C...READ AND PRINT DATA FOR ARRAYS USED IN A 3-D FINITE-DIFFERENCE      GD00040*
C   GROUND-WATER FLOW MODEL...                                          GD00050*
C                                                                       GD00060*
C...DATA CAN BE READ FROM A CARD-IMAGE FILE (80-COLUMN, FORMATTED,      GD00070*
C   SEQUENTIAL) OR UNFORMATTED FILE (SEQUENTIAL).                       GD00080*
C   UPDATES FOR SELECTED CELLS CAN BE READ FROM A CARD-IMAGE FILE.      GD00090*
C   DATA CAN BE PRINTED, WRITTEN ON A CARD-IMAGE FILE OR UNFORMATTED    GD00100*
C   FILE.                                                               GD00110*
C   ALL OPERATIONS ARE PERFORMED ON SELECTED LAYERS IN A MODEL.         GD00120*
C                                                                       GD00130*
C-----------------------------------------------------------------------GD00140*
C                                                                       GD00150*
      INTEGER  RD2, PR2, PR3, FDBAS                                     GD00160*
C                                                                       GD00170*
      DIMENSION  A(1), V(8),                                            GD00180*
     1  FMTI(4), FMTIA(20), FMTO(11), FMTOA(55), FMTP(4), FMTPA(20),    GD00190*
     2  SIDENT(2), SETNAM(2), SETDSK(2), PROJID(2), OPER(4)             GD00200*
C                                                                       GD00210*
      DATA  IEOF, INIT, NSET, NR, NC, NL, KDBAS, KRUP, MXARRY / 9*0 /   GD00220*
     1,     IDISK, BLNK, OPER /0, 1H , 4HMULT, 3HADD, 4HSTOR, 4HSTOR /  GD00230*
C...FILE NUMBERS FOR CARD READER, PRINTER, DATA BASE DISK, AND          GD00240*
C   SEQUENTIAL DISK FILES.                                              GD00250*
     2,     RD2, PR2, PR3, FDBAS / 22, 6, 21, 20 /                      GD00260*
C...INPUT-FILE FORMATS ARE 20F4.0 AND 8F10.0.                           GD00270*
      DATA   FMTIA / 4H(20F,4H4.0),2*1H , 4H(8F1,4H0.0),2*1H ,          GD00280*
     4             12*1H  /                                             GD00290*
C...PRINT FORMATS ARE 18(1X,F6.0), 10(1X,F11.2), 10(1X,F11.8), AND      GD00300*
C   1P10E12.4.                                                          GD00310*
     5,     FMTOA / 4H(1H0,4H,I3,,3H18(,4H1X,F,3H6.0,3H)/(,             GD00320*
     6                    3H4X,,3H18(,4H1X,F,3H6.0,3H))),               GD00330*
     7             4H(1H0,4H,I3,,3H10(,4H1X,F,4H11.2,3H)/(,             GD00340*
     8                    3H4X,,3H10(,4H1X,F,4H11.2,3H))),              GD00350*
     9             4H(1H0,4H,I3,,3H10(,4H1X,F,4H11.8,3H)/(,             GD00360*
     X                    3H4X,,3H10(,4H1X,F,4H11.8,3H))),              GD00370*
     1             4H(1H0,4H,I3,,3H1P,,3H10E,4H12.4,2H/(,               GD00380*
     2                    3H4X,,3H1P,,3H10E,4H12.4,2H)),                GD00390*
     3             11*1H  /                                             GD00400*
C...OUTPUT-FILE FORMATS ARE 20F4.0, 8F10.0, 8F10.3, 8F10.7, AND 1P8E10.3GD00410*
     4,     FMTPA / 4H(20F,4H4.0),2*1H ,                                GD00420*
     5             4H(8F1,4H0.0),2*1H ,                                 GD00430*
     6             4H(8F1,4H0.3),2*1H ,                                 GD00440*
     7             4H(8F1,4H0.7),2*1H ,                                 GD00450*
     8             4H(1P,,4H8E10,3H.3),1H /                             GD00460*
C                                                                       GD00470*
C---------------------------------------------------------------------  GD00480*
C                                                                       GD00490*
C...NOW WHERE DO I GO?                                                  GD00500*
      IF(IOPTN.EQ.1) GO TO 300                                          GD00510*
C                                                                       GD00520*
C-----------------------------------------------------------------------GD00530*
C     ENTRY DBREC -- PROCESS DATA BASE RECORDS WHEN IOPTN=0.            GD00540*
C----------------------------                                           GD00550*
```

```
C
C...HAS THIS ROUTINE BEEN INITIALIZED?
      IF(INIT.EQ.0) GO TO 906
C
C...INITIALIZE VARIABLES.
      II=NI
      IF (II.LE.0) II=1
      JJ=NJ
      IF (JJ.LE.0) JJ=1
      K1=NK1
      IF (K1.LE.0) K1=1
      K2=NK2
      IF (K2.LE.0) K2=1
      NIJ=II*JJ
      I2=NIJ-II
      INC=II
      NN=NIJ
      IF (IATYPE.NE.1) GO TO 2
        NIJ=0
        I2=JJ-1
        INC=1
        NN=JJ
    2 CONTINUE
C
C...COMPARE ARRAY DIMENSIONS WITH MODEL DIMENSIONS FOR DATA BASE.
      IF((IATYPE.EQ.1 .OR. IATYPE.EQ.4) .AND. NN.GT.MXSIZA) GO TO 903
      IF((IATYPE.EQ.2 .OR. IATYPE.EQ.3) .AND. (II.GT.NROW .OR.
     1   JJ.GT.NCOL)) GO TO 903
      IF((IATYPE.EQ.3 .OR. IATYPE.EQ.4) .AND. (K1.GT.K2 .OR.
     1    K2.GT.NLAY)) GO TO 903
C
C-----------------------------
C START LOOP 0.0
C------------------
      DO 200 K=K1,K2
        I3=(K-1)*NIJ
        IF(IPRNTT.LE.0) GO TO 26
          SETNAM(1)=SIDENT(1)
          SETNAM(2)=SIDENT(2)
          LAYR=K
          IRFMT=0
          IRDSK=0
          IFACTA=0
          IUPROW=0
          IUPNOD=0
          IPRNT=IPRNTT
          IWFMT=0
          IWDSK=0
          FACTA=0.
          GO TO 27
   26   READ(RD2,8000) SETNAM, LAYR, IRFMT, IRDSK, IDUM, IFACTA,
     1    IUPROW, IUPNOD, IPRNT, IWFMT, IWDSK, IDUM, IDUM, FACTA
C
            IF(SETNAM(1).NE.SIDENT(1) .OR. SETNAM(2).NE.SIDENT(2)
     1         .OR. LAYR.NE.K) GO TO 911
```

```
C
   27   JJ1=I3+1
        JJ2=I3+NN
        WRITE (PR2,9109) SETNAM, LAYR
C
        IF(IPRNTT.GT.0) GO TO 90
        IRD=0
        IRFMT=IABS(IRFMT)
        IF(IRFMT.EQ.0) GO TO 60
C
C...READ DATA FOR LAYER K FROM A FORMATTED CARD-IMAGE FILE (SEQUENTIAL).
   30 IF(IRFMT.GT.2) GO TO 50
      IRD=IRDSK
      IF(IRD.LE.0) IRD=RD2
      IF(IRD.NE.RD2 .AND. IRD.LE.IFILE) GO TO 912
        WRITE (PR2,9111) IRD
        J2=IRFMT*4
        J1=J2-3
        N=0
        DO 31 J=J1,J2
          N=N+1
   31   FMTI(N)=FMTIA(J)
C
      IF(IRD.EQ.RD2) GO TO 35
        READ(IRD,8003) NL, SETDSK
          IF(SETDSK(1).NE.SETNAM(1) .OR. SETDSK(2).NE.SETNAM(2) .OR.
     1       NL.NE.LAYR) GO TO 911
C
   35   DO 40 I=1,II
          J1=I+I3
          J2=J1+I2
          READ (IRD,FMTI) (A(J), J=J1,J2,INC)
   40   CONTINUE
        GO TO 60
C
C...READ DATA FOR LAYER K FROM AN UNFORMATTED DISK FILE (SEQUENTIAL)..
   50 IF(IRFMT.NE.8) GO TO 55
        IRD=IRDSK
        WRITE (PR2,9112) IRD
        IF(IRD.LE.IFILE) GO TO 912
        READ (IRD) SETDSK, STIM2, NL, NR, NC, (A(J), J=JJ1,JJ2)
          STIME=STIM2
        GO TO 58
C
C...READ DATA FOR LAYER K FROM DATA BASE DISK (RANDOM-ACCESS FILE).
   55 IF(IRFMT.NE.9) GO TO 60
        WRITE (PR2,9113) FDBAS
        GO TO 905
C
   58   IF(SETDSK(1).NE.SETNAM(1) .OR. SETDSK(2).NE.SETNAM(2)
     1    .OR. NL.NE.LAYR .OR. NR.NE.II .OR. NC.NE.JJ) GO TO 911
C
C...ASSIGN THE VALUE FACTA TO ALL CELLS IN LAYER K.
   60 IF(IFACTA.NE.2 .AND. (IRD.NE.0 .OR. IUPDAT.NE.0)) GO TO 64
        WRITE (PR2,9110) FACTA
```

```
          DO 62 J=JJ1,JJ2                                        GD01660*
   62     A(J)=FACTA                                             GD01670*
            FACTA=0.                                             GD01680*
C                                                                GD01690*
C...READ UPDATES FOR SELECTED MODEL ROWS FROM A CARD-IMAGE FILE.  GD01700*
   64   IF (IUPROW.LE.0) GO TO 70                                GD01710*
          WRITE (PR2,9114) RD2                                   GD01720*
          IF (IEOF.GT.0) GO TO 907                               GD01730*
C                                                                GD01740*
   65     READ (RD2,8001,END=75) KLAY, IR, JC1, JC2, IOPCOD, IDUM,   GD01750*
     1         (V(J), J=1,5)                                     GD01760*
            IF ((KLAY*IR*JC1*JC2).LE.0) GO TO 70                 GD01770*
            IF (KLAY.NE.K .OR. IR.GT.II .OR. JC1.GT.JC2 .OR. JC2.GT.JJ)GD01780*
     1         GO TO 908                                         GD01790*
            KN=IABS(IOPCOD)                                      GD01800*
            IF(KN.GT.3) KN=3                                     GD01810*
            IF(KN.EQ.0 .AND. V(1).EQ.0.) V(1)=1.                 GD01820*
            VAL=V(1)                                             GD01830*
            JN=1                                                 GD01840*
            IF(KN.EQ.3) JN=JC2-JC1+1                             GD01850*
            N=KN+1                                               GD01860*
            IN=MIN0(JN,5)                                        GD01870*
            WRITE (PR2,9121) IR, JC1, JC2, OPER(N), (V(J), J=1,IN)   GD01880*
              N=IN                                               GD01890*
C                                                                GD01900*
   67         J2=I3 + IR + (JC1-1)*II                            GD01910*
              IF (INC.EQ.1) J2=JC1                               GD01920*
            DO 68 J=1,IN                                         GD01930*
              IF(KN.EQ.0) A(J2)=A(J2)*VAL                        GD01940*
              IF(KN.EQ.1) A(J2)=A(J2)+VAL                        GD01950*
              IF(KN.EQ.2) A(J2)=VAL                              GD01960*
              IF(KN.EQ.3) A(J2)=V(J)                             GD01970*
   68         J2=J2+INC                                          GD01980*
C                                                                GD01990*
            IF(N.EQ.JN) GO TO 69                                 GD02000*
              JC1=JC1+IN                                         GD02010*
              IN=8                                               GD02020*
              IF(N+IN.GT.JN) IN=JN-N                             GD02030*
              N=N+IN                                             GD02040*
              READ(RD2,8004,END=907) (V(J), J=1,IN)              GD02050*
              WRITE(PR2,9135) (V(J), J=1,IN)                     GD02060*
              GO TO 67                                           GD02070*
   69     GO TO 65                                               GD02080*
C                                                                GD02090*
C...READ UPDATES FOR INDIVIDUAL MODEL CELLS FROM A CARD-IMAGE FILE.   GD02100*
   70   IF (IUPNOD.LE.0) GO TO 80                                GD02110*
          WRITE (PR2,9115) RD2                                   GD02120*
        IF (IEOF.GT.0) GO TO 907                                 GD02130*
C                                                                GD02140*
   71     READ (RD2,8002,END=75) KN, IN, JN, IDUM, VAL           GD02150*
            IF ((KN*IN*JN).LE.0) GO TO 80                        GD02160*
            IF (KN.NE.K .OR. IN.GT.II .OR. JN.GT.JJ) GO TO 908   GD02170*
            J=I3 + IN + (JN-1)*II                                GD02180*
            IF (INC.EQ.1) J=JN                                   GD02190*
            A(J)=VAL                                             GD02200*
```

```
          WRITE (PR2,9122) IN, JN, A(J)
         GO TO 71
C
  75       IEOF=1
C
C...MULTIPLY FACTA TIMES ALL VALUES IN THIS LAYER.
  80 IF (IFACTA.NE.0 .OR. FACTA.EQ.0. .OR. FACTA.EQ.1.) GO TO 85
        WRITE (PR2,9116) FACTA
        DO 81 J=JJ1,JJ2
  81    A(J)=A(J)*FACTA
C
C
C...ADD FACTA TO ALL VALUES IN THIS LAYER.
  85 IF(IFACTA.NE.1 .OR. FACTA.EQ.0.) GO TO 90
         WRITE(PR2,9130) FACTA
       DO 86 J=JJ1,JJ2
  86 A(J)=A(J)+FACTA
C
C...PRINT DATA FOR LAYER K.
  90    IF (IPRNT.LE.0) GO TO 100
          WRITE (PR2,9117) PR3
          WRITE (PR3,9124) SETNAM, K, STIME
          IF (IPRNT.GT.4) IPRNT=4
          J2=IPRNT*11
          J1=J2-10
          N=0
          DO 91 J=J1,J2
            N=N+1
  91      FMTO(N)=FMTOA(J)
C
          DO 95 I=1,II
            J1=I+I3
            J2=J1+I2
  95      WRITE (PR3,FMTO) I, (A(J), J=J1,J2,INC)
C
C...WRITE DATA FOR LAYER K ON FORMATTED DISK FILE (SEQUENTIAL).
 100    IF (IWFMT.LE.0) GO TO 200
        IF (IWFMT.GT.5) GO TO 110
          WRITE (PR2,9118) IWDSK
          IF(IWDSK.LE.IFILE) GO TO 912
          J2=IWFMT*4
          J1=J2-3
          N=0
          DO 105 J=J1,J2
            N=N+1
 105      FMTP(N)=FMTPA(J)
C
          WRITE (IWDSK,8003) LAYR, SETNAM
C
          DO 108 I=1,II
            J1=I+I3
            J2=J1+I2
 108      WRITE (IWDSK,FMTP) (A(J), J=J1,J2,INC)
C
C...WRITE DATA FOR LAYER K ON UNFORMATTED DISK FILE (SEQUENTIAL).
```

```
  110   IF (IWFMT.NE.8) GO TO 120                                         GD02760*
          WRITE (PR2,9119) IWDSK                                          GD02770*
          IF(IWDSK.LE.IFILE) GO TO 912                                    GD02780*
          WRITE (IWDSK) SETNAM, STIME, LAYR, II, JJ, (A(J), J=JJ1,JJ2)   GD02790*
C                                                                         GD02800*
C...WRITE DATA FOR LAYER K ON DATA BASE DISK (RANDOM-ACCESS FILE).        GD02810*
  120   IF(IWFMT.NE.9) GO TO 200                                          GD02820*
          WRITE (PR2,9120) FDBAS                                          GD02830*
          GO TO 905                                                       GD02840*
C                                                                         GD02850*
  200 CONTINUE                                                            GD02860*
C                                                                         GD02870*
C------------------                                                       GD02880*
C END LOOP 0.0                                                            GD02890*
C---------------------------                                              GD02900*
C                                                                         GD02910*
  299 GO TO 999                                                           GD02920*
C                                                                         GD02930*
C----------------------------------------------------------------------   GD02940*
C     ENTRY DBOPEN -- OPEN DATA BASE FILE WHEN IOPTN=1.                   GD02950*
C-----------------------------                                            GD02960*
C...INITIALIZE SUBROUTINE GWDBAS...                                       GD02970*
  300   NROW=NI                                                           GD02980*
        NCOL=NJ                                                           GD02990*
        NLAY=NK2                                                          GD03000*
        MXARRY=NROW*NCOL                                                  GD03010*
        MXSIZA=MAX0(NROW, NCOL, NLAY)                                     GD03020*
        IFILE=MAX0(RD2, PR2, PR3, FDBAS)                                  GD03030*
        INIT=1                                                            GD03040*
      READ(RD2,8000) PROJID, IDISK                                        GD03050*
        IDISK=0                                                           GD03060*
      WRITE(PR2,9127) PROJID, IDISK, NLAY, NROW, NCOL                     GD03070*
      GO TO 999                                                           GD03080*
C                                                                         GD03090*
C-------------------------------------------------------------------------GD03100*
C                                                                         GD03110*
C...ERROR MESSAGES.                                                       GD03120*
  900 IMESS=900                                                           GD03130*
      GO TO 998                                                           GD03140*
  901 IMESS=901                                                           GD03150*
      GO TO 998                                                           GD03160*
  902 IMESS=902                                                           GD03170*
      GO TO 998                                                           GD03180*
  903 IMESS=903                                                           GD03190*
      GO TO 998                                                           GD03200*
  904 IMESS=904                                                           GD03210*
      GO TO 998                                                           GD03220*
  905 IMESS=905                                                           GD03230*
      GO TO 998                                                           GD03240*
  906 IMESS=906                                                           GD03250*
      GO TO 998                                                           GD03260*
  907 IMESS=907                                                           GD03270*
      GO TO 998                                                           GD03280*
  908 IMESS=908                                                           GD03290*
      GO TO 998                                                           GD03300*
```

```
  909 IMESS=909                                                         GD03310*
      GO TO 998                                                         GD03320*
  910 IMESS=910                                                         GD03330*
      GO TO 998                                                         GD03340*
  911 IMESS=911                                                         GD03350*
      GO TO 998                                                         GD03360*
  912 IMESS=912                                                         GD03370*
      GO TO 998                                                         GD03380*
  913 IMESS=913                                                         GD03390*
      GO TO 998                                                         GD03400*
  914 IMESS=914                                                         GD03410*
      GO TO 998                                                         GD03420*
  915 IMESS=915                                                         GD03430*
      GO TO 998                                                         GD03440*
  916 IMESS=916                                                         GD03450*
      GO TO 998                                                         GD03460*
  917 IMESS=917                                                         GD03470*
C                                                                       GD03480*
  998 WRITE (PR2,9190) IMESS                                            GD03490*
      STOP                                                              GD03500*
C                                                                       GD03510*
  999 RETURN                                                            GD03520*
C                                                                       GD03530*
C-----------------------------------------------------------------------GD03540*
C                                                                       GD03550*
 8000 FORMAT(2A4,2X, 12I5, F10.0)                                       GD03560*
 8001 FORMAT(6I5, 5F10.0)                                               GD03570*
 8002 FORMAT(4I5, F10.0)                                                GD03580*
 8003 FORMAT(I2, 2A4)                                                   GD03590*
 8004 FORMAT(8F10.0)                                                    GD03600*
 9109 FORMAT( '-INITIALIZE CELLS FOR DATA SET = ',2A4,  ', LAYER = ', GD03610*
     1 I2, 1H: )                                                        GD03620*
 9110 FORMAT(1H0,3X,  'SET ALL CELLS IN LAYER TO CONSTANT = ',1P,E12.5)GD03630*
 9111 FORMAT(1H0,3X,  'READ FROM CARD-IMAGE FILE (FILE NO. ', I2,      GD03640*
     1 ') VALUES FOR ALL CELLS IN LAYER.')                              GD03650*
 9112 FORMAT(1H0,3X,  'READ FROM SEQUENTIAL UNFORMATTED DISK FILE',    GD03660*
     1 ' (FILE NO. ',I2,  ') VALUES FOR ALL CELLS IN LAYER.')          GD03670*
 9113 FORMAT(1H0,3X,  'READ FROM DATA BASE DISK FILE (FILE NO. ',      GD03680*
     1 I2,  ') VALUES FOR ALL CELLS IN LAYER.')                         GD03690*
 9114 FORMAT(1H0,3X,  'READ FROM CARD-IMAGE FILE (FILE NO. ',I2,       GD03700*
     1  ') UPDATES FOR SELECTED ROWS IN LAYER.' /                       GD03710*
     2 1H0,5X,  'ROW  START COL.  END COL.  OPERATION  ...VALUES...'/) GD03720*
 9115 FORMAT(1H0,3X,  'READ FROM CARD-IMAGE FILE (FILE NO. ',I2,       GD03730*
     1 ') UPDATES FOR SELECTED CELLS IN LAYER.' /                       GD03740*
     2 1H0,5X,  'ROW  COL   VALUE' /)                                   GD03750*
 9116 FORMAT(1H0,3X,  'MULTIPLY VALUES FOR ALL CELLS IN LAYER TIMES CONGD03760*
     1STANT = ', 1P,E12.5)                                              GD03770*
 9117 FORMAT(1H0,3X,  'PRINT OUT (FILE NO. ',I2,  ') VALUES FOR ALL CEGD03780*
     1LLS IN LAYER.')                                                   GD03790*
 9118 FORMAT(1H0,3X,  'WRITE ON CARD-IMAGE FILE (FILE NO. ',I2,        GD03800*
     1 ') VALUES FOR ALL CELLS IN LAYER.')                              GD03810*
 9119 FORMAT(1H0,3X,  'WRITE ON SEQUENTIAL UNFORMATTED DISK FILE',     GD03820*
     1 ' (FILE NO. ',I2, ') VALUES FOR ALL CELLS IN LAYER.')           GD03830*
 9120 FORMAT(1H0,3X,  'WRITE ON DATA BASE DISK FILE (FILE NO. ',I2,    GD03840*
     1 ') VALUES FOR ALL CELLS IN LAYER.')                              GD03850*
```

```
9121 FORMAT(6X, I3,5X, I3,8X, I3,7X, A4,3X, 1P,5E14.5)                  GD03860*
9122 FORMAT(6X, I3,2X, I3,2X, 1P,E12.5)                                   GD03870*
9123 FORMAT(1H-, 35HNEW RECORDS ADDED TO DATA BASE DISK, 3H = , I5 /   GD03880*
   1  1X, 35HDATA BASE RECORDS UPDATED THIS TIME, 3H = , I5 /             GD03890*
   1  1X, 31HTOTAL RECORDS ON DATA BASE DISK,4X, 3H = , I5 /              GD03900*
   2  1X, 31HSPACE AVAILABLE FOR NEW RECORDS,4X, 3H = , I5 )              GD03910*
9124 FORMAT(1H1,20X, 22HVALUES FOR DATA SET = ,2A4, 10H, LAYER = , I2, GD03920*
   1  20H, SIMULATION TIME = , F12.4 / 21X, 74(1H-) / 4H0ROW)             GD03930*
9125 FORMAT(  '-IDENTIFICATION RECORD FOR THE DATA BASE DISK FILE',     GD03940*
   1  ' (FILE NO. ',I2, 2H). /                                            GD03950*
   2 1H0,5X, 18HPROJECT IDENTIFIER, 33X, 3H = , 2A4 /                     GD03960*
   3 6X, 36HMAX. RECORDS SPECIFIED FOR DATA BASE, 15X, 3H = ,I8 /         GD03970*
   4 6X, 50HMAX. RECORD LENGTH (BYTES) SPECIFIED FOR DATA BASE, 1X,       GD03980*
   4         3H = ,I8 /                                                   GD03990*
   5 6X, 17HNO. ROWS IN MODEL, 34X, 3H = ,I8 /                            GD04000*
   6 6X, 20HNO. COLUMNS IN MODEL,31X, 3H = ,I8 /                          GD04010*
   7 6X, 19HNO. LAYERS IN MODEL, 32X, 3H = ,I8 /                          GD04020*
   8 6X, 22HNO. DATA SETS IN MODEL,29X, 3H = ,I8 )                        GD04030*
9126 FORMAT(1H-,5X,   '---SUMMARY OF DATA SETS IN DATA BASE---' /         GD04040*
   1 1H0,18X,  'NAME       LAYERS' / 19X,  '--------  ------' /           GD04050*
   2  (19X, 2A4, 2X, I4) )                                                GD04060*
9127 FORMAT(1H-,56(1H-)/ 1X, 'INITIALIZE SUBROUTINE GWDBAS FOR PROCESSIGD04070*
   1NG GROUP V DATA'/ 1X, 56(1H-) /                                       GD04080*
   2 1H0,5X,  'PROJECT IDENTIFIER',12X, 3H = , 2A4 /                      GD04090*
   3 1H0,5X,  'ACCESS CODE FOR DATA BASE DISK', 3H = , I5 /               GD04100*
   4 1H0,5X,  'MODEL DIMENSIONS:  LAYERS      ', 3H = , I5 /              GD04110*
   5 1H0,5X,  '                   ROWS        ', 3H = , I5 /              GD04120*
   6 1H0,5X,  '                   COLUMNS     ', 3H = , I5 )              GD04130*
9128 FORMAT(1H-,5X,  'OPEN DATA BASE ON DISK FILE NO. ',I2)               GD04140*
9129 FORMAT(1H0,5X,  'DATA BASE DISK FILE WAS NOT ACCESSED.')             GD04150*
9130 FORMAT(1H0,3X,  'ADD TO VALUES FOR ALL CELLS IN THIS LAYER A ',      GD04160*
   1  'CONSTANT = ', 1P,E12.5)                                            GD04170*
9131 FORMAT('-*** SUMMARY OF ACTIVITY FOR THE DATA BASE DISK ACCESSED',GD04180*
   1  ' IN SUBR. GWDBAS ***')                                              GD04190*
9132 FORMAT('-*** END OF PROCESSING FOR GROUP V DATA IN SUBROUTINE GWDBGD04200*
   1AS ***'/)                                                             GD04210*
9133 FORMAT( '-NEW IDENTIFICATION RECORD WRITTEN ON DATA BASE DISK.')  GD04220*
9134 FORMAT('-DATA BASE DISK IS CLOSED.')                                 GD04230*
9135 FORMAT(42X, 1P,5E14.5)                                               GD04240*
9190 FORMAT(  '-*** ERROR MESSAGE NO. ',I3,  '  FROM SUBR. GWDBAS ***'/GD04250*
   1    '-*** JOB TERMINATED ***' //)                                     GD04260*
     END                                                                  GD04270*
```

```
      SUBROUTINE MSQDIF (HCAL, T, DELX, DELY, SETID, STIME)
C
C...COMPUTE THE ROOT-MEAN-SQUARE FOR DIFFERENCES BETWEEN OBSERVED
C   VALUES OF A MODEL PARAMETER AND MODEL-COMPUTED VALUES.
C
C     HCAL = COMPUTED VALUES OF A PARAMETER.  COORDINATES ARE REFERENCED
C            TO A SUBREGIONAL MODEL GRID.
C     SETID= ALPHANUMERIC IDENTIFIER FOR THE DATA SET OF COMPUTED VALUES
C     STIME= SIMULATION TIME FOR A GROUNDWATER-FLOW MODEL.
C     HOBS = OBSERVED VALUES OF A PARAMETER.  COORDINATES ARE REFERENCED
C            TO A REGIONAL MODEL GRID.
C     KNT  = NO. OF OBSERVED VALUES FOR A MODEL PARAMETER.
C     AVE  = SUMT/KNT, WHERE SUMT IS THE SUM OF (HOBS-HCAL).
C     RMSE = SQRT(SUMT2/KNT), WHERE SUMT2 IS THE SUM OF (HOBS-HCAL)**2.
C
      REAL*8  HCAL, SUMT, SUMT2, SUML, SUML2, RN, DBLE, DSQRT
      DIMENSION  HCAL(1), T(1), DELX(1), DELY(1), SETID(2), SETID2(2)
C
      INTEGER  RD, RD2, RD3, RD4, PR
      DATA  RD, RD2, RD3, RD4, PR / 41, 42, 43, 44, 6 /
     1,     INIT / 0 /
C
C...CHECK IF GROUP VI DATA HAVE BEEN READ.
      IF(INIT.GT.0) GO TO 10
        INIT=1
        WRITE(PR,9001)
C
C...READ SPECIFICATIONS FOR THE REGIONAL MODEL.
      READ(RD,8000,END=901) NROW, NCOL, NLAY
        WRITE(PR,9003) NROW, NCOL, NLAY
        NN=NROW*NCOL*NLAY
        IF(NN.LE.0) GO TO 903
C
C...READ SPECIFICATIONS FOR BOUNDARIES OF SUBREGIONAL MODEL.
      READ(RD,8000,END=901) NRA1, NRA2, NCA1, NCA2
        IF(NRA1.LE.0) NRA1=1
        IF(NRA2.LE.0) NRA2=NROW
        IF(NCA1.LE.0) NCA1=1
        IF(NCA2.LE.0) NCA2=NCOL
C
      WRITE(PR,9004) NRA1, NRA2, NCA1, NCA2
C
      IF(NRA1.GT.NRA2 .OR. NRA2.GT.NROW) GO TO 904
      IF(NCA1.GT.NCA2 .OR. NCA2.GT.NCOL) GO TO 904
C
      NRA=NRA2-NRA1+1
      NCA=NCA2-NCA1+1
      NR1=NRA-1
      NC1=NCA-1
      NNA=NRA*NCA
      IF(NNA.LE.0) GO TO 904
C
      GO TO 999
C
C...READ COMPUTED VALUES FOR ALL CELLS IN A 3-D SUBREGIONAL GROUNDWATER
```

```
C   FLOW MODEL.  NOTE THAT COORDINATES ARE REFERENCED TO THE SUBREGIONAL
C   MODEL GRID.
   10 CONTINUE
      WRITE(PR,9000)
C-    READ(RD3,END=905) SETID STIME
      WRITE(PR,9002) SETID, STIME
C-    READ(RD3,END=905) (HCAL(J), J=1,NNA)
C-    READ(RD3,END=905) (T(J), J=1,NNA)
C-    READ(RD3,END=905) (DELX(J), J=1,NCA)
C-    READ(RD3,END=905) (DELY(J), J=1,NRA)
C
C...INITIALIZE VARIABLES.
      SUMT=0.
      SUMT2=0.
      KNT=0
      DIFMIN= 9.E70
      DIFMAX=0.
C
C...READ OBSERVED VALUES FOR SELECTED CELLS IN A 3-D REGIONAL GROUNDWATE
C   FLOW MODEL.  NOTE THAT MODEL COORDINATES ARE REFERENCED TO THE
C   REGIONAL MODEL GRID.
C
   20 READ(RD2,8004,END=38) K, SETID2
      WRITE(PR,9006) SETID2, K, RD2, K
      IF(K.LE.0 .OR. K.GT.NLAY) GO TO 906
        SUML=0.
        SUML2=0.
        KNTL=0
        DIFMNL= 9.E70
        DIFMXL=0.
C
   21 READ(RD2,*,END=907) I, J, DI, DJ, HOBS
        IF(I.LE.0) GO TO 31
        IF(I.GT.NROW .OR. J.LE.0 .OR. J.GT.NCOL) GO TO 908
        IF(I.LT.NRA1 .OR. I.GT.NRA2) GO TO 30
        IF(J.LT.NCA1 .OR. J.GT.NCA2) GO TO 30
C
        II=I-NRA1+1
        JJ=J-NCA1+1
        N=II + (JJ-1)*NRA + (K-1)*NNA
        IF(T(N).LE.0.) GO TO 30
C
C...USE INTERPOLATION TO DETERMINE COMPUTED HEAD AT THE SAME COORDINATES
C   AS THE OBSERVED HEAD.
      H=HCAL(N)
      IF(DI.EQ.0. .AND. DJ.EQ.0.) GO TO 24
      IF((DI.LT.0. .AND. II.LE.2) .OR. (DI.GT.0 .AND. II.GE.NR1) .OR.
     1   (DJ.LT.0. .AND. JJ.LE.2) .OR. (DJ.GT.0. .AND. JJ.GE.NC1))
     2   GO TO 22
      IF (DI) 12, 13, 14
   12 DY=DELY(II-1)
      HY=HCAL(N-1)
      GO TO 15
   13 DY=1.
      HY=H
```

```
      GO TO 15
   14 DY=DELY(II+1)
      HY=HCAL(N+1)
   15 IF (DJ) 16, 17, 18
   16 DX=DELX(JJ-1)
      HX=HCAL(N-NRA)
      GO TO 19
   17 DX=1.
      HX=H
      GO TO 19
   18 DX=DELX(JJ+1)
      HX=HCAL(N+NRA)
   19 H=H + (HX-H)*DJ*2./(DX+DELX(JJ)) + (HY-H)*DI*2./(DY+DELY(II))
      GO TO 24
   22 DI=0.
      DJ=0.
C
   24   DIF=H-HOBS
        WRITE(PR,9007) II, JJ, I, J, DI, DJ, HOBS, H, DIF
C
        SUML=SUML+DIF
        SUML2=SUML2+DBLE(DIF*DIF)
        KNTL=KNTL+1
C
        IF(ABS(DIF).LE.ABS(DIFMXL)) GO TO 26
          DIFMXL=DIF
          IMAXL=II
          JMAXL=JJ
   26   IF(ABS(DIF).GE.ABS(DIFMNL)) GO TO 30
          DIFMNL=DIF
          IMINL=II
          JMINL=JJ
   30 GO TO 21
C
   31 IF(KNTL.GT.0) GO TO 32
        WRITE(PR,9008) K
        GO TO 35
C
   32 RN=KNTL
      AVE=SUML/RN
      RMSE=DSQRT(SUML2/RN)
      WRITE(PR,9012) K, RMSE, AVE, KNTL, DIFMXL, K, IMAXL, JMAXL,
     1  DIFMNL, K, IMINL, JMINL
C
      SUMT=SUMT+SUML
      SUMT2=SUMT2+SUML2
      KNT=KNT+KNTL
C
      IF(ABS(DIFMXL).LE.ABS(DIFMAX)) GO TO 33
        DIFMAX=DIFMXL
        IMAX=IMAXL
        JMAX=JMAXL
        KMAX=K
   33 IF(ABS(DIFMNL).GE.ABS(DIFMIN)) GO TO 35
        DIFMIN=DIFMNL
```

```
      IMIN=IMINL                                                        MS01660*
      JMIN=JMINL                                                        MS01670*
      KMIN=K                                                            MS01680*
C                                                                       MS01690*
   35 GO TO 20                                                          MS01700*
C                                                                       MS01710*
   38 IF(KNT.EQ.0) GO TO 909                                            MS01720*
      RN=KNT                                                            MS01730*
      AVE=SUMT/RN                                                       MS01740*
      RMSE=DSQRT(SUMT2/RN)                                              MS01750*
      WRITE(PR,9014) RMSE, AVE, KNT, DIFMAX, KMAX, IMAX, JMAX, DIFMIN,  MS01760*
     1   KMIN, IMIN, JMIN                                               MS01770*
C                                                                       MS01780*
      GO TO 998                                                         MS01790*
C                                                                       MS01800*
C                                                                       MS01810*
  901 J=901                                                             MS01820*
        GO TO 950                                                       MS01830*
  902 J=902                                                             MS01840*
        GO TO 950                                                       MS01850*
  903 J=903                                                             MS01860*
        GO TO 950                                                       MS01870*
  904 J=904                                                             MS01880*
        GO TO 950                                                       MS01890*
  905 J=905                                                             MS01900*
        GO TO 950                                                       MS01910*
  906 J=906                                                             MS01920*
        GO TO 950                                                       MS01930*
  907 J=907                                                             MS01940*
        GO TO 950                                                       MS01950*
  908 J=908                                                             MS01960*
        GO TO 950                                                       MS01970*
  909 J=909                                                             MS01980*
        GO TO 950                                                       MS01990*
  910 J=910                                                             MS02000*
  950 WRITE(PR,9950) J                                                  MS02010*
      STOP                                                              MS02020*
  998 WRITE(PR,9990)                                                    MS02030*
  999 RETURN                                                            MS02040*
C                                                                       MS02050*
C                                                                       MS02060*
 8000 FORMAT(5I5)                                                       MS02070*
 8004 FORMAT(I2, 2A4)                                                   MS02080*
 9000 FORMAT(1H1, 'COMPUTATION OF ROOT-MEAN-SQUARE FOR DIFFERENCES BETWEMS02090*
     1EN OBSERVED AND MODEL-COMPUTED HEAD' / 1X, 88(1H-) )              MS02100*
 9001 FORMAT(1H-, 'DATA FROM RECORD 1 IN GROUP VI DATA: PARAMETERS FOR CMS02110*
     1OMPUTING ROOT-MEAN-SQUARE OF DIFFERENCES BETWEEN OBSERVED AND COMPMS02120*
     2UTED HEAD' / 1X, 128(1H-) )                                       MS02130*
 9002 FORMAT(  '-COMPUTED VALUES ARE FOR MODEL PARAMETER = ',2A4,       MS02140*
     1  ', SIMULATION TIME = ', F14.2)                                  MS02150*
 9003 FORMAT(1H0, 34HSPECIFICATIONS FOR REGIONAL MODEL:/                MS02160*
     1 1H0,5X, 10HMODEL ROWS,30X, 3H = ,I5/                             MS02170*
     2      6X, 13HMODEL COLUMNS,27X, 3H = ,I5/                         MS02180*
     3      6X, 12HMODEL LAYERS,28X, 3H = ,I5 )                         MS02190*
 9004 FORMAT(38H-SPECIFICATIONS FOR SUBREGIONAL MODEL:/                 MS02200*
```

```
     2 1H0,5X, 9HFIRST ROW,31X, 3H = ,I5/
     3 6X, 8HLAST ROW,32X, 3H = ,I5/
     4 6X, 12HFIRST COLUMN,28X,3H = ,I5/
     5 6X, 11HLAST COLUMN,29X,3H = ,I5)
 9006 FORMAT(// '-OBSERVED VALUES FOR MODEL PARAMETER:' /
     1 6X, 13HDATA SET NAME,2X, 3H = ,2A4 /
     2 6X, 11HMODEL LAYER,4X, 3H = ,I5 /
     3 6X, 10HINPUT FILE,5X, 3H = ,I5 /
     3 1H-,3X, 'OBSERVED AND COMPUTED VALUES FOR MODEL LAYER ', I2,
     3 1H: //
     4 6X, ' MODEL     REGIONAL    ....DISPLACEMENT....     OBSERVED
     4  COMPUTED' /
     5 6X, 'ROW  COL   ROW  COL       Y             X         VALUES
     5   VALUES    DIFFERENCE' )
 9007 FORMAT(4X, 2(2I5,1X), 2(F12.1,1X), 3(F12.3,1X) )
 9008 FORMAT(1H0,7X, 'NO OBSERVATIONS FOR LAYER ',I2)
 9012 FORMAT(1H-,3X, 'STATISTICAL SUMMARY FOR MODEL LAYER ',I2,1H: //
     1 6X, 32HROOT-MEAN SQUARE FOR DIFFERENCES,3X, 3H = ,F12.3 /
     2 6X, 19HMEAN OF DIFFERENCES,16X, 3H = ,F12.3 /
     3 6X, 19HNO. OF OBSERVATIONS,16X, 3H = ,I8 /
     4 6X, 15HMAX. DIFFERENCE,20X, 3H = ,F12.3, 25H  FOR (LAYER,ROW,COL)
     5 = (, I2,1H,, I3,1H,, I3, 1H) /
     6 6X, 15HMIN. DIFFERENCE,20X, 3H = ,F12.3, 25H  FOR (LAYER,ROW,COL)
     7 = (, I2,1H,, I3,1H,, I3,1H) )
 9014 FORMAT(/1H-,3X, 'STATISTICAL SUMMARY FOR ALL MODEL LAYERS:' /
     1 6X, 32HROOT-MEAN SQUARE FOR DIFFERENCES,3X, 3H = ,F12.3 /
     2 6X, 19HMEAN OF DIFFERENCES,16X, 3H = ,F12.3 /
     3 6X, 19HNO. OF OBSERVATIONS,16X, 3H = ,I8 /
     4 6X, 15HMAX. DIFFERENCE,20X, 3H = ,F12.3, 25H  FOR (LAYER,ROW,COL)
     5 = (, I2,1H,, I3,1H,, I3, 1H) /
     6 6X, 15HMIN. DIFFERENCE,20X, 3H = ,F12.3, 25H  FOR (LAYER,ROW,COL)
     7 = (, I2,1H,, I3,1H,, I3,1H) )
 9950 FORMAT( '-*** ERROR CODE IN SUBR. MSQDIF = ', I3, 4H *** /
     1  '-*** JOB IS TERMINATED ***' //)
 9990 FORMAT( '-*** END OF PROCESSING IN SUBR. MSQDIF ***'//)
      END
```

```
      SUBROUTINE PRNTAI(PHI,STRT,T,S,WELL,DELX,DELY)
C     ------------------------------------------------------------------
C     PRINT MAPS OF DRAWDOWN AND HYDRAULIC HEAD
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI,Z,XLABEL,YLABEL,TITLE,XN1,MESUR
      REAL *4K
C
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), S(I0,J0,K0), WELL(I0,J0,K
     10), DELX(J0), DELY(I0), T(I0,J0,K0)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,       NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1
     1, ERMULT, STEADY, DELT1
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANK
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2
C
C...DEFINE PRINT FILE.
      DATA  IPR / 9 /
      GO TO 999
C     ..................................................................
C
C     ---INITIALIZE VARIABLES FOR PLOT---
C     ********************
      ENTRY MAP
C     ********************
      YDIM=0.
      WIDTH=0.
      DO 10 J=2,J1
   10 WIDTH=WIDTH+DELX(J)
      DO 20 I=2,I1
   20 YDIM=YDIM+DELY(I)
   30 XSF=DINCH*XSCALE
      YSF=DINCH*YSCALE
      NYD=YDIM/YSF
      IF (NYD*YSF.LE.YDIM-DELY(I1)/2.) NYD=NYD+1
      IF (NYD.LE.12) GO TO 40
      DINCH=YDIM/(12.*YSCALE)
      WRITE (6,330) DINCH
      IF (YSCALE.LT.1.0) WRITE (6,340)
      GO TO 30
   40 NXD=WIDTH/XSF
      IF (NXD*XSF.LE.WIDTH-DELX(J1)/2.) NXD=NXD+1
      N4=NXD*N1+1
      N5=NXD+1
      N6=NYD+1
      N8=N2*NYD+1
      NA(1)=N4/2-1
```

```
      NA(2)=N4/2                                                        PR00560
      NA(3)=N4/2+3                                                      PR00570
      NC=(N3-N8-10)/2                                                   PR00580
      ND=NC+N8                                                          PR00590
      NE=MAX0(N5,N6)                                                    PR00600
      VF1(3)=DIGIT(ND)                                                  PR00610
      VF2(3)=DIGIT(ND)                                                  PR00620
      VF3(3)=DIGIT(NC)                                                  PR00630
      XLABEL(3)=MESUR                                                   PR00640
      YLABEL(6)=MESUR                                                   PR00650
      DO 60 I=1,NE                                                      PR00660
      NNX=N5-I                                                          PR00670
      NNY=I-1                                                           PR00680
      IF (NNY.GE.N6) GO TO 50                                           PR00690
      YN(I)=YSF*NNY/YSCALE                                              PR00700
   50 IF (NNX.LT.0) GO TO 60                                            PR00710
      XN(I)=XSF*NNX/YSCALE                                              PR00720
   60 CONTINUE                                                          PR00730
      GO TO 999                                                         PR00740*
C     ..................................................................PR00750
C                                                                       PR00760
C     ********************                                              PR00770
      ENTRY PRNTA(NG,LA)                                                PR00780
C     ********************                                              PR00790
C     ---VARIABLES INITIALIZED EACH TIME A PLOT IS REQUESTED---         PR00800
      DIST=WIDTH-DELX(J1)/2.                                            PR00810
      JJ=J1                                                             PR00820
      LL=1                                                              PR00830
      Z=NXD*XSF                                                         PR00840
      IF (NG.EQ.1) WRITE (IPR,300) (TITLE(I),I=1,3),LA,KP,SUM           PR00850*
      IF (NG.EQ.2) WRITE (IPR,300) (TITLE(I),I=4,6),LA,KP,SUM           PR00860*
      DO 290 I=1,N4                                                     PR00870
C                                                                       PR00880
C     ---LOCATE X AXES---                                               PR00890
      IF (I.EQ.1.OR.I.EQ.N4) GO TO 70                                   PR00900
      PRNT(1)=SYM(12)                                                   PR00910
      PRNT(N8)=SYM(12)                                                  PR00920
      IF ((I-1)/N1*N1.NE.I-1) GO TO 90                                  PR00930
      PRNT(1)=SYM(14)                                                   PR00940
      PRNT(N8)=SYM(14)                                                  PR00950
      GO TO 90                                                          PR00960
C                                                                       PR00970
C     ---LOCATE Y AXES---                                               PR00980
   70 DO 80 J=1,N8                                                      PR00990
      IF ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)                           PR01000
   80 IF ((J-1)/N2*N2.NE.J-1) PRNT(J)=SYM(13)                           PR01010
C                                                                       PR01020
C     ---COMPUTE LOCATION OF CELLS AND DETERMINE APPROPRIATE SYMBOL---  PR01030
   90 IF (DIST.LT.0..OR.DIST.LT.Z-XN1*XSF) GO TO 240                    PR01040
      YLEN=DELY(2)/2.                                                   PR01050
      DO 220 L=2,I1                                                     PR01060
      J=YLEN*N2/YSF+1.5                                                 PR01070
      IF (T(L,JJ,LA).EQ.0.) GO TO 160                                   PR01080
      IF (S(L,JJ,LA).LT.0.) GO TO 210                                   PR01090
      INDX3=0                                                           PR01100
```

```
      GO TO (100,110), NG                                                PR01110
  100 K=(STRT(L,JJ,LA)-PHI(L,JJ,LA))*FACT1                               PR01120
C     -TO CYCLE SYMBOLS FOR DRAWDOWN, REMOVE C FROM COL. 1 OF NEXT CARD-PR01130
C     K=AMOD(K,10.)                                                      PR01140
      GO TO 120                                                          PR01150
  110 K=PHI(L,JJ,LA)*FACT2                                               PR01160
  120 IF (K) 130,160,140                                                 PR01170
  130 IF (J-2.GT.0) PRNT(J-2)=SYM(13)                                    PR01180
      N=-K+.5                                                            PR01190
      IF (N.LT.100) GO TO 150                                            PR01200
      GO TO 190                                                          PR01210
  140 N=K+.5                                                             PR01220
      IF (N.LT.100) GO TO 150                                            PR01230
      IF (N.GT.999) GO TO 190                                            PR01240
      INDX3=N/100                                                        PR01250
      IF (J-2.GT.0) PRNT(J-2)=SYM(INDX3)                                 PR01260
      N=N-INDX3*100                                                      PR01270
  150 INDX1=MOD(N,10)                                                    PR01280
      IF (INDX1.EQ.0) INDX1=10                                           PR01290
C     -TO CYCLE SYMBOLS FOR DRAWDOWN, REMOVE C FROM COL. 1 OF NEXT CARD-PR01300
C     IF (NG.EQ.1) GO TO 170                                             PR01310
      INDX2=N/10                                                         PR01320
      IF (INDX2.GT.0) GO TO 180                                          PR01330
      INDX2=10                                                           PR01340
      IF (INDX3.EQ.0) INDX2=15                                           PR01350
      GO TO 180                                                          PR01360
  160 INDX1=15                                                           PR01370
  170 INDX2=15                                                           PR01380
  180 IF (J-1.GT.0) PRNT(J-1)=SYM(INDX2)                                 PR01390
      PRNT(J)=SYM(INDX1)                                                 PR01400
      GO TO 220                                                          PR01410
  190 DO 200 II=1,3                                                      PR01420
      JI=J-3+II                                                          PR01430
  200 IF (JI.GT.0) PRNT(JI)=SYM(11)                                      PR01440
  210 IF (S(L,JJ,LA).LT.0.) PRNT(J)=SYM(16)                              PR01450
  220 YLEN=YLEN+(DELY(L)+DELY(L+1))/2.                                   PR01460
  230 DIST=DIST-(DELX(JJ)+DELX(JJ-1))/2.                                 PR01470
      JJ=JJ-1                                                            PR01480
      IF (JJ.EQ.0) GO TO 240                                             PR01490
      IF (DIST.GT.Z-XN1*XSF) GO TO 230                                   PR01500
  240 CONTINUE                                                           PR01510
C                                                                        PR01520
C     ---PRINT AXES,LABELS, AND SYMBOLS---                               PR01530
      IF (I-NA(LL).EQ.0) GO TO 260                                       PR01540
      IF ((I-1)/N1*N1-(I-1)) 270,250,270                                 PR01550
  250 WRITE (IPR,VF1) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XN(1+(I-1)/6)   PR01560*
      GO TO 280                                                          PR01570
  260 WRITE (IPR,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XLABEL(LL)      PR01580*
      LL=LL+1                                                            PR01590
      GO TO 280                                                          PR01600
  270 WRITE (IPR,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8)                 PR01610*
C                                                                        PR01620
C     ---COMPUTE NEW VALUE FOR Z AND INITIALIZE PRNT---                  PR01630
  280 Z=Z-2.*XN1*XSF                                                     PR01640
      DO 290 J=1,N8                                                      PR01650
```

```
  290 PRNT(J)=SYM(15)                                                   PR01660
C                                                                       PR01670
C     ---NUMBER AND LABEL Y AXIS AND PRINT LEGEND---                    PR01680
      WRITE (IPR,VF3) (BLANK(J),J=1,NC),(YN(I),I=1,N6)                  PR01690*
      WRITE (IPR,320) (YLABEL(I),I=1,6)                                 PR01700*
      IF (NG.EQ.1) WRITE (IPR,310) FACT1                                PR01710*
      IF (NG.EQ.2) WRITE (IPR,310) FACT2                                PR01720*
  999 RETURN                                                            PR01730*
C                                                                       PR01740
C                                                                       PR01750
  300 FORMAT (1H1,20X, 3A8, ', LAYER = ',I3, ', STRESS PERIOD = ',I4,   PR01760*
     1 ', SIMULATION TIME = ',F14.2 / 21X, 93(1H-)/ )                   PR01770*
  310 FORMAT ('0EXPLANATION'/' ',11('-')//' R = CONSTANT HEAD BOUNDARY'/PR01780
     1' *** = VALUE EXCEEDED 3 FIGURES'/' MULTIPLICATION FACTOR =',F8.3)PR01790
  320 FORMAT ('0',39X,6A8)                                              PR01800
  330 FORMAT ('0',25X,10('*'),' TO FIT MAP WITHIN 12 INCHES, DINCH REVISPR01810
     1ED TO',G15.7,1X,10('*'))                                          PR01820
  340 FORMAT ('0',45X,'NOTE: GENERALLY SCALE SHOULD BE > OR = 1.0')     PR01830
      END                                                               PR01840
```

```
      SUBROUTINE SOLVE(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FAC
     1T,EL,FL,GL,V,XI,TEST3,QRE,LAYRCH,IDR,RH,RC,RB,IDN,DRCOF,DRELV,ISEA
     2 )
C     ------------------------------------------------------------------
C     SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL*8 PHI,RHO,B,D,F,H,Z,SU,RHOP,W,WMIN,RHO1,RHO2,RHO3,XPART,YPART
     1,ZPART,DMIN1,WMAX,XT,YT,ZT,DABS,DMAX1,DEN,TXM,TYM,TZM,QR,ZERO,ONE
     2,E,AL,BL,CL,A,C,G,WU,TU,U,DL,RES,SUPH,GLXI,ZPHI,UXR,UXD,ELN,FLN,
     3 GLN,VN,DTEST2
C
      DIMENSION PHI(1), STRT(1), OLD(1), T(1), S(1), TR(1), TC(1), TK(1)
     1, WELL(1), DELX(1),DELY(1),DELZ(1),FACT(1,3,K0),  RHOP(20), TEST3(
     21), EL(1), FL(1), GL(1), V(1), XI(1), QRE(1), LAYRCH(1)
     3,IDR(1),RH(1),RC(1),RB(1),IDN(1),DRCOF(1),DRELV(1),ISEA(1)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,      NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1
     1, ERMULT, STEADY, DELT1
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)
C
C...DEFINE PARAMETERS FOR THE IBM LIBRARY ROUTINE ERRSET.
      COMMON/ERDAT1/ LOC, KLOC, ILOC, JLOC
      EXTERNAL  ERMES1
      DATA IERNO, IERNO2, IERCNT, IERMSG, ITRACE / 207, 209, 100, 10, 2/
C
C...DEFINE FLOATING-POINT CONSTANTS USED IN THIS ROUTINE.
      DATA ZERO, ONE, DTEST2 / 0.D0, 1.D0, 1.D-20 /
C
C...DEFINE PRINT FILE FOR HEAD AFTER EACH ITERATION.
      DATA  IPR /10/
C
C...MAXIMUM NO. OF WORDS IN THE ARRAY RHOP.
      DATA  MXRHOP / 20 /
      GO TO 999
C     ..................................................................
C     **********************
      ENTRY ITER
C     **********************
      IF(LENGTH.GT.MXRHOP) GO TO 901
      CALL ERRSET (IERNO, IERCNT, IERMSG, ITRACE, ERMES1, IERNO2)
      LOC=1
      KLOC=0
      ILOC=0
      JLOC=0
C     ---COMPUTE AND PRINT ITERATION PARAMETERS---
      WMIN=ONE
      P2=LENGTH-1
```

```
      IF(WMAX1.LE.0.) GO TO 2                                           SL00560*
        WMAX=WMAX1                                                      SL00570*
        GO TO 48                                                        SL00580*
    2 XT=3.141593**2/(2.*J2*J2)                                         SL00590*
      YT=3.141593**2/(2.*I2*I2)                                         SL00600
      ZT=3.141593**2/(2.*K0*K0)                                         SL00610
      RHO1=ZERO                                                         SL00620*
      RHO2=ZERO                                                         SL00630*
      RHO3=ZERO                                                         SL00640*
      DO 40 K=1,K0                                                      SL00650
        DZ=DELZ(K)                                                      SL00660*
        NIA=(K-1)*NIJ                                                   SL00670*
      DO 40 J=2,J1                                                      SL00680
        DX=DELX(J)                                                      SL00690*
        NIB=(J-1)*I0+NIA                                                SL00700*
      DO 40 I=2,I1                                                      SL00710*
        N=I+NIB                                                         SL00720*
      IF(T(N).LE.0.) GO TO 40                                           SL00730
      D=TR(N-I0)/DX                                                     SL00740*
      F=TR(N)/DX                                                        SL00750*
      B=TC(N-1)/DELY(I)                                                 SL00760
      H=TC(N)/DELY(I)                                                   SL00770
      SU=ZERO                                                           SL00780*
      Z=ZERO                                                            SL00790*
      IF (K.NE.1 .AND. T(N-NIJ).GT.0.) Z=TK(N-NIJ)                      SL00800*
      IF (K.NE.K0 .AND. T(N+NIJ).GT.0.) SU=TK(N)                        SL00810*
      IF(IEQN.NE.ICHK(11)) GO TO 10                                     SL00820*
        Z=Z/DZ                                                          SL00830*
        SU=SU/DZ                                                        SL00840*
   10 CONTINUE                                                          SL00850
      TXM=DMAX1(D,F)                                                    SL00860
      TYM=DMAX1(B,H)                                                    SL00870
      TZM=DMAX1(SU,Z)                                                   SL00880
      DEN=DMIN1(D,F)                                                    SL00890
      IF (DEN.EQ.ZERO) DEN=TXM                                          SL00900*
      IF (DEN.EQ.ZERO) GO TO 20                                         SL00910*
      RHO1=DMAX1(RHO1,TYM/DEN)                                          SL00920
   20 DEN=DMIN1(B,H)                                                    SL00930
      IF (DEN.EQ.ZERO) DEN=TYM                                          SL00940*
      IF (DEN.EQ.ZERO) GO TO 30                                         SL00950*
      RHO2=DMAX1(RHO2,TXM/DEN)                                          SL00960
   30 DEN=DMIN1(SU,Z)                                                   SL00970
      IF (DEN.EQ.ZERO) DEN=TZM                                          SL00980*
      IF (DEN.EQ.ZERO) GO TO 40                                         SL00990*
      RHO3=DMAX1(RHO3,TXM/DEN)                                          SL01000
   40 CONTINUE                                                          SL01010
      XPART=XT/(ONE+RHO1)                                               SL01020*
      YPART=YT/(ONE+RHO2)                                               SL01030*
      ZPART=ZT/(ONE+RHO3)                                               SL01040*
      WMIN=DMIN1(WMIN,XPART,YPART,ZPART)                                SL01050
      WMAX=ONE-WMIN                                                     SL01060*
   48 PJ=-1.                                                            SL01070*
      B=ONE-WMAX                                                        SL01080*
      DO 50 I=1,LENGTH                                                  SL01090
      PJ=PJ+1.                                                          SL01100
```

```
   50 RHOP(I)=ONE-B**(PJ/P2)                                      SL01110*
      WRITE (6,9001) LENGTH,(RHOP(J),J=1,LENGTH)                  SL01120*
      GO TO 998                                                   SL01130*
C     ............................................................SL01140
C                                                                 SL01150
C     ---INITIALIZE DATA FOR A NEW ITERATION---                   SL01160
   60 IT=IT+1                                                     SL01170
      IF (IT.LE.ITMAX) GO TO 70                                   SL01180
      IERR=2                                                      SL01190*
      IT2=IT                                                      SL01200*
      IT=IT-1                                                     SL01210*
      GO TO 230                                                   SL01220*
C                                                                 SL01230*
   70 IF (MOD(IT,LENGTH)) 80,80,90                                SL01240
C     **********************                                      SL01250
      ENTRY NEWITA                                                SL01260
C     **********************                                      SL01270
      IFLAG=0                                                     SL01280*
      IF(ISALT.GT.0) IFLAG=1                                      SL01290*
C                                                                 SL01300*
C...DETERMINE RECHARGE LAYERS FOR WT CELLS THAT ARE DRY.          SL01310*
      IF(IWATER.NE.ICHK(6) .OR. IQRE.NE.ICHK(7) .OR. LRECH.LE.0)  SL01320*
     +    GO TO 80                                                SL01330*
      DO 76 J=2,J1                                                SL01340*
        II=(J-1)*I0                                               SL01350*
      DO 76 I=2,I1                                                SL01360*
        NXY=II+I                                                  SL01370*
        LAYRCH(NXY)=0                                             SL01380*
        K=K0                                                      SL01390*
      DO 75 KK=LRECH,K0                                           SL01400*
        N=(K-1)*NIJ+NXY                                           SL01410*
        IF(T(N).LE.0.) GO TO 75                                   SL01420*
        IF(S(N).GE.0.) LAYRCH(NXY)=K                              SL01430*
        GO TO 76                                                  SL01440*
   75 K=K-1                                                       SL01450*
   76 CONTINUE                                                    SL01460*
C                                                                 SL01470*
   80 NTH=0                                                       SL01480
   90 NTH=NTH+1                                                   SL01490
      W=RHOP(NTH)                                                 SL01500
      IT2=IT+1                                                    SL01510*
      TEST3(IT2)=0.                                               SL01520*
      BIGH=0.                                                     SL01530*
      IB=0                                                        SL01540*
      JB=0                                                        SL01550*
      KB=0                                                        SL01560*
      TEST=0.0                                                    SL01570
      BIG=0.                                                      SL01580
C                                                                 SL01590*
      DO 100 I=1,NNOD                                             SL01600*
      EL(I)=0.                                                    SL01610
      FL(I)=0.                                                    SL01620
      GL(I)=0.                                                    SL01630
      V(I)=0.                                                     SL01640
  100 XI(I)=0.                                                    SL01650
```

```
C                                                                       SL01660
      IF(IT.EQ.0) CALL ERRSET (IERNO, IERCNT, IERMSG, ITRACE,           SL01670*
     1   ERMES1, IERNO2)                                                SL01680*
      LOC=20                                                            SL01690*
      KLOC=0                                                            SL01700*
      ILOC=0                                                            SL01710*
      JLOC=0                                                            SL01720*
C                                                                       SL01730*
C   ---COMPUTE TRANSMISSIVITY AND T-COEFFICIENTS FOR MODEL CELLS---     SL01740*
      IF(IWATER.EQ.ICHK(6)) CALL TRANS                                  SL01750*
      IF(IFLAG.GT.0) CALL TCOF                                          SL01760*
      IFLAG=0                                                           SL01770*
C                                                                       SL01780*
C     ---CHOOSE SIP NORMAL OR REVERSE ALGORITHM---                      SL01790
  110 IF (MOD(IT,2)) 120,120,170                                        SL01800
C                                                                       SL01810*
C     ---BEGIN SIP NORMAL ALGORITHM---                                  SL01820*
  120 DO 150 K=1,K0                                                     SL01830
        L1=(K-1)*NIJ                                                    SL01840*
        DZ=DELZ(K)                                                      SL01850*
        KLOC=K                                                          SL01860*
      DO 150 I=2,I1                                                     SL01870
        DY=DELY(I)                                                      SL01880*
        ILOC=I                                                          SL01890*
      DO 150 J=2,J1                                                     SL01900
        NXY=(J-1)*I0+I                                                  SL01910*
        N=NXY+L1                                                        SL01920*
C                                                                       SL01930
C     ---SKIP COMPUTATIONS IF CELL OUTSIDE MODEL---                     SL01940
      IF (T(N).LE.0. .OR. S(N).LT.0.) GO TO 150                         SL01950
C                                                                       SL01960
      NIA=N+1                                                           SL01970
      NIB=N-1                                                           SL01980
      NJA=N+I0                                                          SL01990
      NJB=N-I0                                                          SL02000
      NKA=N+NIJ                                                         SL02010
      NKB=N-NIJ                                                         SL02020
      DX=DELX(J)                                                        SL02030*
      LOC=21                                                            SL02040*
      JLOC=J                                                            SL02050*
C                                                                       SL02060
C     ---COMPUTE COEFFICIENTS---                                        SL02070
      D=TR(NJB)/DX                                                      SL02080*
      F=TR(N)/DX                                                        SL02090*
      B=TC(NIB)/DY                                                      SL02100*
      H=TC(N)/DY                                                        SL02110*
      SU=ZERO                                                           SL02120*
      Z=ZERO                                                            SL02130*
      IF(K.NE.1 .AND. T(NKB).GT.0.) Z=TK(NKB)                           SL02140*
      IF(K.NE.K0 .AND. T(NKA).GT.0.) SU=TK(N)                           SL02150*
      IF(IEQN.NE.ICHK(11)) GO TO 125                                    SL02160*
        Z=Z/DZ                                                          SL02170*
        SU=SU/DZ                                                        SL02180*
  125 RHO=S(N)/DELT                                                     SL02190*
      QR=ZERO                                                           SL02200*
```

```
      UXR=ZERO                                                          SL02210*
      UXD=ZERO                                                          SL02220*
      PHIX=PHI(N)                                                       SL02230*
C                                                                       SL02240*
      IF (IQRE.EQ.ICHK(7) .AND. K.EQ.LAYRCH(NXY)) QR=QRE(NXY)           SL02250*
C                                                                       SL02260*
      IF (NRIV.LE.0 .OR. IDR(N).LE.0) GO TO 130                         SL02270*
      ND=IDR(N)                                                         SL02280*
      IF(ISALT.LE.0 .OR. ISEA(N).LE.0) GO TO 128                        SL02290*
      IF(ISEA(N).EQ.2 .AND. T(NKA).GT.0.) GO TO 130                     SL02300*
C-    IF(PHIX.LE.RH(ND)) GO TO 130                                      SL02310*
  128 X=RB(ND)                                                          SL02320*
      UXR=RC(ND)                                                        SL02330*
      IF(PHIX.GT.X) X=0.                                                SL02340*
      QR=QR+UXR*(RH(ND)-X)                                              SL02350*
      IF(PHIX.LE.RB(ND)) UXR=ZERO                                       SL02360*
C                                                                       SL02370*
  130 IF (NDRAIN.LE.0 .OR. IDN(N).LE.0) GO TO 135                       SL02380*
      ND=IDN(N)                                                         SL02390*
      UXD=DRCOF(ND)                                                     SL02400*
      IF (PHIX.LE.DRELV(ND)) UXD=ZERO                                   SL02410*
      QR=QR+UXD*DRELV(ND)                                               SL02420*
C                                                                       SL02430*
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---         SL02440
  135 E=-B-D-F-H-SU-Z-RHO-UXR-UXD                                       SL02450*
        LOC=22                                                          SL02460*
      BL=B/(1.+W*(EL(NIB)+GL(NIB)))                                     SL02470
      CL=D/(1.+W*(FL(NJB)+GL(NJB)))                                     SL02480
      C=BL*EL(NIB)                                                      SL02490
      G=CL*FL(NJB)                                                      SL02500
      WU=CL*GL(NJB)                                                     SL02510
      U=BL*GL(NIB)                                                      SL02520
      IF (K.EQ.1) GO TO 140                                             SL02530
C                                                                       SL02540*
      LOC=23                                                            SL02550*
      AL=Z/(1.+W*(EL(NKB)+FL(NKB)))                                     SL02560
      A=AL*EL(NKB)                                                      SL02570
      TU=AL*FL(NKB)                                                     SL02580
        LOC=24                                                          SL02590*
      DL=E+W*(A+C+G+WU+TU+U)-CL*EL(NJB)-BL*FL(NIB)-AL*GL(NKB)           SL02600
      IF(DABS(DL).LE.DTEST2) GO TO 148                                  SL02610*
      EL(N)=(F-W*(A+C))/DL                                              SL02620
      FL(N)=(H-W*(G+TU))/DL                                             SL02630
      GL(N)=(SU-W*(WU+U))/DL                                            SL02640
      SUPH=ZERO                                                         SL02650*
      IF (K.NE.K0) SUPH=SU*PHI(NKA)                                     SL02660
      LOC=25                                                            SL02670*
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-Z*PSL02680
     1HI(NKB)-WELL(N)-RHO*OLD(N)-QR                                     SL02690
      LOC=26                                                            SL02700*
      V(N)=(RES*BETA1-AL*V(NKB)-BL*V(NIB)-CL*V(NJB))/DL                 SL02710*
      GO TO 150                                                         SL02720
C                                                                       SL02730*
  140 LOC=27                                                            SL02740*
      DL=E+W*(C+G+WU+U)-CL*EL(NJB)-BL*FL(NIB)                           SL02750
```

```
      IF(DABS(DL).LE.DTEST2) GO TO 148                                  SL02760*
      EL(N)=(F-W*C)/DL                                                  SL02770
      FL(N)=(H-W*G)/DL                                                  SL02780
      GL(N)=(SU-W*(WU+U))/DL                                            SL02790
      SUPH=ZERO                                                         SL02800*
      IF (K.NE.K0) SUPH=SU*PHI(NKA)                                     SL02810
      LOC=28                                                            SL02820*
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-WELSL02830
     1L(N)-RHO*OLD(N)-QR                                                SL02840
      LOC=29                                                            SL02850*
      V(N)=(RES*BETA1-BL*V(NIB)-CL*V(NJB))/DL                           SL02860*
        GO TO 150                                                       SL02870*
  148 WRITE(6,9006) IT, LOC, I, J, K, DL, E                             SL02880*
      T(N)=0.                                                           SL02890*
      EL(N)=0.                                                          SL02900*
      FL(N)=0.                                                          SL02910*
      GL(N)=0.                                                          SL02920*
      V(N)=0.                                                           SL02930*
      XI(N)=0.                                                          SL02940*
      IFLAG=1                                                           SL02950*
  150 CONTINUE                                                          SL02960
C                                                                       SL02970
C     ---BACK SUBSTITUTE FOR VECTOR XI---                               SL02980
      LOC=30                                                            SL02990*
      DO 160 K=1,K0                                                     SL03000
      K3=K0-K+1                                                         SL03010
      KLOC=K3                                                           SL03020*
      NIA=(K3-1)*NIJ                                                    SL03030*
      DO 160 I=1,I2                                                     SL03040
      I3=I0-I                                                           SL03050
      ILOC=I3                                                           SL03060*
      NIB=I3+NIA                                                        SL03070*
      DO 160 J=1,J2                                                     SL03080
      J3=J0-J                                                           SL03090
      N=(J3-1)*I0+NIB                                                   SL03100*
      IF (T(N).LE.0. .OR. S(N).LT.0.) GO TO 160                         SL03110
      JLOC=J3                                                           SL03120*
      GLXI=ZERO                                                         SL03130*
      IF (K3.NE.K0) GLXI=GL(N)*XI(N+NIJ)                                SL03140
      DEL=V(N)-EL(N)*XI(N+I0)-FL(N)*XI(N+1)-GLXI                        SL03150*
      PHI(N)=PHI(N)+DEL                                                 SL03160*
      XI(N)=DEL                                                         SL03170*
C                                                                       SL03180
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERIA---           SL03190
      X=ABS(DEL)                                                        SL03200*
      IF(X.LE.BIG) GO TO 160                                            SL03210*
        BIG=X                                                           SL03220*
        BIGH=DEL                                                        SL03230*
        IB=I3                                                           SL03240*
        JB=J3                                                           SL03250*
        KB=K3                                                           SL03260*
  160 CONTINUE                                                          SL03270
C                                                                       SL03280*
      GO TO 215                                                         SL03290*
C                                                                       SL03300*
```

```
C     ..............................................................SL03310
C     ---BEGIN SIP REVERSE ALGORITHM---                              SL03320*
  170 DO 200 KK=1,K0                                                 SL03330
      K=K0-KK+1                                                      SL03340
      L1=(K-1)*NIJ                                                   SL03350*
      DZ=DELZ(K)                                                     SL03360*
      KLOC=K                                                         SL03370*
      DO 200 II=1,I2                                                 SL03380
      I=I0-II                                                        SL03390
      DY=DELY(I)                                                     SL03400*
      ILOC=I                                                         SL03410*
      DO 200 J=2,J1                                                  SL03420
      NXY=(J-1)*I0+I                                                 SL03430*
      N=NXY+L1                                                       SL03440*
C                                                                    SL03450
C     ---SKIP COMPUTATIONS IF CELL OUTSIDE AQUIFER---                SL03460
      IF (T(N).LE.0. .OR. S(N).LT.0.) GO TO 200                      SL03470
C                                                                    SL03480
      NIA=N+1                                                        SL03490
      NIB=N-1                                                        SL03500
      NJA=N+I0                                                       SL03510
      NJB=N-I0                                                       SL03520
      NKA=N+NIJ                                                      SL03530
      NKB=N-NIJ                                                      SL03540
      DX=DELX(J)                                                     SL03550*
      LOC=41                                                         SL03560*
      JLOC=J                                                         SL03570*
C                                                                    SL03580
C     ---COMPUTE COEFFICIENTS---                                     SL03590
      D=TR(NJB)/DX                                                   SL03600*
      F=TR(N)/DX                                                     SL03610*
      B=TC(NIB)/DY                                                   SL03620*
      H=TC(N)/DY                                                     SL03630*
      SU=ZERO                                                        SL03640*
      Z=ZERO                                                         SL03650*
      IF(K.NE.1 .AND. T(NKB).GT.0.) Z=TK(NKB)                        SL03660*
      IF(K.NE.K0 .AND. T(NKA).GT.0.) SU=TK(N)                        SL03670*
      IF(IEQN.NE.ICHK(11)) GO TO 175                                 SL03680*
        Z=Z/DZ                                                       SL03690*
        SU=SU/DZ                                                     SL03700*
  175 RHO=S(N)/DELT                                                  SL03710*
      QR=ZERO                                                        SL03720*
      UXR=ZERO                                                       SL03730*
      UXD=ZERO                                                       SL03740*
      PHIX=PHI(N)                                                    SL03750*
C                                                                    SL03760*
      IF (IQRE.EQ.ICHK(7) .AND. K.EQ.LAYRCH(NXY)) QR=QRE(NXY)        SL03770*
C                                                                    SL03780*
      IF (NRIV.LE.0 .OR. IDR(N).LE.0) GO TO 180                      SL03790*
      ND=IDR(N)                                                      SL03800*
      IF(ISALT.LE.0 .OR. ISEA(N).LE.0) GO TO 178                     SL03810*
      IF(ISEA(N).EQ.2 .AND. T(NKA).GT.0.) GO TO 180                  SL03820*
C-    IF(PHIX.LE.RH(ND)) GO TO 180                                   SL03830*
  178 X=RB(ND)                                                       SL03840*
      UXR=RC(ND)                                                     SL03850*
```

```
      IF(PHIX.GT.X) X=0.                                                SL03860*
      QR=QR+UXR*(RH(ND)-X)                                              SL03870*
      IF(PHIX.LE.RB(ND)) UXR=ZERO                                       SL03880*
C                                                                       SL03890*
  180 IF (NDRAIN.LE.0 .OR. IDN(N).LE.0) GO TO 185                       SL03900*
      ND=IDN(N)                                                         SL03910*
      UXD=DRCOF(ND)                                                     SL03920*
      IF(PHIX.LE.DRELV(ND)) UXD=ZERO                                    SL03930*
      QR=QR+UXD*DRELV(ND)                                               SL03940*
C                                                                       SL03950
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---         SL03960
  185 E=-B-D-F-H-SU-Z-RHO-UXR-UXD                                       SL03970*
        LOC=42                                                          SL03980*
      BL=H/(1.+W*(EL(NIA)+GL(NIA)))                                     SL03990
      CL=D/(1.+W*(FL(NJB)+GL(NJB)))                                     SL04000
      C=BL*EL(NIA)                                                      SL04010
      G=CL*FL(NJB)                                                      SL04020
      WU=CL*GL(NJB)                                                     SL04030
      U=BL*GL(NIA)                                                      SL04040
      IF (K.EQ.K0) GO TO 190                                            SL04050
C                                                                       SL04060*
      LOC=43                                                            SL04070*
      AL=SU/(1.+W*(EL(NKA)+FL(NKA)))                                    SL04080
      A=AL*EL(NKA)                                                      SL04090
      TU=AL*FL(NKA)                                                     SL04100
      LOC=44                                                            SL04110*
      DL=E+W*(C+G+A+WU+TU+U)-AL*GL(NKA)-BL*FL(NIA)-CL*EL(NJB)           SL04120
      IF(DABS(DL).LE.DTEST2) GO TO 198                                  SL04130*
      EL(N)=(F-W*(C+A))/DL                                              SL04140
      FL(N)=(B-W*(G+TU))/DL                                             SL04150
      GL(N)=(Z-W*(WU+U))/DL                                             SL04160
      ZPHI=ZERO                                                         SL04170*
      IF (K.NE.1) ZPHI=Z*PHI(NKB)                                       SL04180
      LOC=45                                                            SL04190*
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SU*PHI(NSL04200
     1KA)-ZPHI-WELL(N)-RHO*OLD(N)-QR                                    SL04210
      LOC=46                                                            SL04220*
      V(N)=(RES*BETA1-AL*V(NKA)-BL*V(NIA)-CL*V(NJB))/DL                 SL04230*
      GO TO 200                                                         SL04240
C                                                                       SL04250*
  190 LOC=47                                                            SL04260*
      DL=E+W*(C+G+WU+U)-BL*FL(NIA)-CL*EL(NJB)                           SL04270
      IF(DABS(DL).LE.DTEST2) GO TO 198                                  SL04280*
      EL(N)=(F-W*C)/DL                                                  SL04290
      FL(N)=(B-W*G)/DL                                                  SL04300
      GL(N)=(Z-W*(WU+U))/DL                                             SL04310
      ZPHI=ZERO                                                         SL04320*
      IF (K.NE.1) ZPHI=Z*PHI(NKB)                                       SL04330
      LOC=48                                                            SL04340*
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-ZPHI-WELSL04350
     1L(N)-RHO*OLD(N)-QR                                                SL04360
      LOC=49                                                            SL04370*
      V(N)=(RES*BETA1-BL*V(NIA)-CL*V(NJB))/DL                           SL04380*
        GO TO 200                                                       SL04390*
  198 WRITE(6,9006) IT, LOC, I, J, K, DL, E                             SL04400*
```

```
      T(N)=0.
      EL(N)=0.
      FL(N)=0.
      GL(N)=0.
      V(N)=0.
      XI(N)=0.
      IFLAG=1
  200 CONTINUE
C
C     ---BACK SUBSTITUTE FOR VECTOR XI---
      LOC=50
      DO 210 K=1,K0
        KLOC=K
        NIA=(K-1)*NIJ
      DO 210 I=2,I1
        ILOC=I
        NIB=I+NIA
      DO 210 J=1,J2
      J3=J0-J
      N=(J3-1)*I0+NIB
      IF (T(N).LE.0. .OR. S(N).LT.0.) GO TO 210
      JLOC=J3
      GLXI=ZERO
      IF (K.NE.1) GLXI=GL(N)*XI(N-NIJ)
      DEL=V(N)-EL(N)*XI(N+I0)-FL(N)*XI(N-1)-GLXI
      PHI(N)=PHI(N)+DEL
      XI(N)=DEL
C
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERIA---
      X=ABS(DEL)
      IF(X.LE.BIG) GO TO 210
        BIG=X
        BIGH=DEL
        IB=I
        JB=J3
        KB=K
  210 CONTINUE
C
C     ---SAVE MAX. HEAD CHANGE FOR EACH ITERATION---
  215 IF (BIG.GT.ERR) TEST=1.
      TEST3(IT2)=BIGH
      NODCH(1,IT2)=IB
      NODCH(2,IT2)=JB
      NODCH(3,IT2)=KB
C
C     ---PRINT COMPUTED HEAD AFTER EACH ITERATION (ITRHED=1) OR
C        PRINT HEAD CHANGE FOR EACH ITERATION (ITRHED=2)---
      IF (ITRHED.EQ.0) GO TO 221
        K3=NIJ-I0
      DO 220 K=1,K0
      IF (LEVEL2(K).EQ.0) GO TO 220
      WRITE (IPR,9004) K, KP, KT, IT
        I3=(K-1)*NIJ
      DO 220 I=1,I0
        J3=I+I3
```

```
          J4=J3+K3                                                      SL04960*
          IF(ITRHED.EQ.1) WRITE(IPR,9005) I, (PHI(J), J=J3,J4,I0)       SL04970*
          IF(ITRHED.EQ.2) WRITE(IPR,9005) I, (XI(J), J=J3,J4,I0)        SL04980*
  220 CONTINUE                                                          SL04990*
  221 CONTINUE                                                          SL05000*
C                                                                       SL05010*
C     ---DO ANOTHER ITERATION?                                          SL05020*
      IF(TEST.NE.0.) GO TO 60                                           SL05030*
C                                                                       SL05040*
C     ---PRINT MAX. HEAD CHANGE FOR EACH ITERATION---                   SL05050*
  230 WRITE(6,9002) KP, KT, IT                                          SL05060*
      WRITE(6,9003) (TEST3(J), (NODCH(I,J),I=1,3), J=1,IT2)             SL05070*
C                                                                       SL05080*
      IF(IERR.GT.0) WRITE(6,9000)                                       SL05090*
C                                                                       SL05100*
  998 CALL ERRSET (IERNO, 10, 10, 2, 1, IERNO2)                         SL05110*
  999 RETURN                                                            SL05120*
C                                                                       SL05130*
  901 WRITE(6,9901) MXRHOP                                              SL05140*
      STOP                                                              SL05150*
C                                                                       SL05160*
C     ..................................................................SL05170*
C                                                                       SL05180*
 9000 FORMAT ( '-*** EXCEEDED MAXIMUM ITERATIONS SPECIFIED FOR EQUATION-SL05190*
     1SOLVING PROCEDURE ***')                                           SL05200*
 9001 FORMAT( '-*** COMPUTATION OF ITERATION PARAMETERS FOR SIP',       SL05210*
     1 ' EQUATION-SOLVING PROCEDURE ***'/                               SL05220*
     2 1H0,4X, 'NO. ITERATION PARAMETERS = ',I3/                        SL05230*
     3  (1H0,4X, 1P,8E16.7) )                                           SL05240*
 9002 FORMAT(92H-MAXIMUM HEAD CHANGE FOR EACH ITERATION IN THE EQUATION-SL05250*
     1SOLVING PROCEDURE.  STRESS PERIOD = ,I4, 14H, TIME STEP = ,I4 /   SL05260*
     2 1X, 113(1H-) // 6X, 17HNO. ITERATIONS = ,I3 //                   SL05270*
     3 4(29H    HEAD CHANGE  CELL (I,J,K)) )                            SL05280*
 9003 FORMAT( 4(3X, F12.2,2X, I3,1H,, I3,1H,, I3,1X) )                  SL05290*
 9004 FORMAT (1H1,20X, 36HCOMPUTED HEAD (OR CHANGE) FOR LAYER ,I2,      SL05300*
     1 19H.  STRESS PERIOD = ,I4, 14H, TIME STEP = ,I4, 14H, ITERATION  SL05310*
     2= ,  I4 / 21X, 97(1H-) / 5H0 ROW )                                SL05320*
 9005 FORMAT(1H0,I4, 12F10.2 / (5X, 12F10.2))                           SL05330*
 9006 FORMAT(  '0***ERROR IN SUBR. SOLVE***  IT = ',I2,   ', LOC = ',I2,SL05340*
     1  ', (I,J,K) = (', I3,1H,, I3,1H,, I3,  '), DL = ',1P,D10.3,      SL05350*
     2  ', E = ',1P,D10.3,  '.  SET T=0.' )                             SL05360*
 9901 FORMAT(  '-***ERROR IN SUBR. SOLVE*** NO. ITERATION PARAMETERS EXCSL05370*
     1EEDS ',I3,  ' (SIZE OF THE ARRAY RHOP).' )                        SL05380*
      END                                                               SL05390
```

```
      SUBROUTINE STEP(PHI,STRT,OLD,T,EL)                                ST00010*
C     ------------------------------------------------------------------ST00020
C     INITIALIZE DATA FOR A NEW TIME STEP AND PRINT RESULTS             ST00030
C     ------------------------------------------------------------------ST00040
C                                                                       ST00050
C     SPECIFICATIONS:                                                   ST00060
      REAL*8  PHI, TFLX, CFLX, CTFLX, SUMFLX                            ST00070*
      REAL*8  XLABEL,YLABEL,TITLE,XN1,MESUR                             ST00080
C                                                                       ST00090
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0ST00100
     1), EL(I0,J0,K0), SETID(6)                                         ST00110*
C                                                                       ST00120
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,       NPER,KTH,ITMAX,LENGTH,KP,NST00130*
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCST00140
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN   ST00150*
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC   ST00160*
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD       ST00170*
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY                               ST00180*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,BETA1,WMAX1      ST00190*
     1, ERMULT, STEADY, DELT1                                           ST00200*
      COMMON /SARRAY/ ICHK(20),LEVEL1(40),LEVEL2(40), NODCH(3,51)       ST00210*
      COMMON /CK/ SUMFLX(4,41), TFLX(9,2), CFLX(9,2,40), CTFLX(9,2),    ST00220*
     1  FLX(9,2,40)                                                     ST00230*
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANKST00240
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),ST00250
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                          ST00260
C                                                                       ST00270*
C...DEFINE NAME OF PLOT-FILE DATA SET.                                  ST00280*
      DATA  SETID /4HPLOT,4HHEAD, 4*1H /                                ST00290*
C                                                                       ST00300*
C...DEFINE PRINT FILE FOR NUMERIC HEAD AND DRAWDOWN.                    ST00310*
      DATA  IPR / 8 /                                                   ST00320*
      GO TO 998                                                         ST00330*
C     ..................................................................ST00340
C     ********************                                              ST00350
      ENTRY NEWSTP                                                      ST00360
C     ********************                                              ST00370
      KT=KT+1                                                           ST00380
      IT=0                                                              ST00390
      DO 10 K=1,K0                                                      ST00400
      DO 10 J=1,J0                                                      ST00410
      DO 10 I=1,I0                                                      ST00420
   10 OLD(I,J,K)=PHI(I,J,K)                                             ST00430
      DELT=CDLT*DELT                                                    ST00440
      SUM=SUM+DELT                                                      ST00450
      SUMP=SUMP+DELT                                                    ST00460
      DAYSP=SUMP/86400.                                                 ST00470
      YRSP=DAYSP/365.                                                   ST00480
      HRS=SUM/3600.                                                     ST00490
      SMIN=HRS*60.                                                      ST00500
      DAYS=HRS/24.                                                      ST00510
      YRS=DAYS/365.                                                     ST00520
      GO TO 998                                                         ST00530*
C                                                                       ST00540
C     ---PRINT OUTPUT AT DESIGNATED TIME STEPS---                       ST00550
```

```
C     **********************                                            ST00560
      ENTRY OUTPUT                                                      ST00570
C     **********************                                            ST00580
C                                                                       ST00590*
C      ---PRINT MASS BALANCE---                                         ST00600*
      IF((IFLO.NE.ICHK(3) .OR. MOD(KT,KTH).NE.0) .AND. IFINAL.NE.2)     ST00610*
     1    GO TO 30                                                      ST00620*
        CALL CWRITE                                                     ST00630*
C                                                                       ST00640*
   30 IF(MOD(KT,KTH).NE.0 .AND. IFINAL.NE.2) GO TO 998                  ST00650*
C                                                                       ST00660*
C...PRINT ALPHANUMERIC MAPS.                                            ST00670*
      IF (XSCALE.LE.0.) GO TO 70                                        ST00680
      IF (FACT1.EQ.0.) GO TO 50                                         ST00690
C     ---PRINT DRAWDOWN MAPS---                                         ST00700*
      DO 40 K=1,K0                                                      ST00710*
        J=LEVEL1(K)                                                     ST00720*
        IF(J.GT.0) CALL PRNTA (1,J)                                     ST00730*
   40 CONTINUE                                                          ST00740*
C                                                                       ST00750*
   50 IF(FACT2.EQ.0.) GO TO 70                                          ST00760
C     ---PRINT HEAD MAPS---                                             ST00770*
      DO 60 K=1,K0                                                      ST00780*
        J=LEVEL2(K)                                                     ST00790*
        IF(J.GT.0) CALL PRNTA (2,J)                                     ST00800*
   60 CONTINUE                                                          ST00810*
C                                                                       ST00820*
C...PRINT NUMERIC TABLES.                                               ST00830*
   70 IF (IDRAW.NE.ICHK(1)) GO TO 100                                   ST00840
C     ---PRINT DRAWDOWN TABLES---                                       ST00850*
      DO 90 K=1,K0                                                      ST00860
        IF (LEVEL1(K).EQ.0) GO TO 90                                    ST00870*
      WRITE (IPR,9006) K, KP, KT, SUM                                   ST00880*
      DO 90 I=1,I0                                                      ST00890
      DO 80 J=1,J0                                                      ST00900
        EL(1,J,1)=0.                                                    ST00910*
        IF(T(I,J,K).GT.0.) EL(1,J,1)=STRT(I,J,K)-SNGL(PHI(I,J,K))       ST00920*
   80 CONTINUE                                                          ST00930*
      WRITE (IPR,9005) I,(EL(1,J,1),J=1,J0)                             ST00940*
   90 CONTINUE                                                          ST00950*
C                                                                       ST00960*
C     ---SET PHI = -99  WHERE T=0---                                    ST00970*
  100 DO 105 K=1,K0                                                     ST00980*
      DO 105 J=1,J0                                                     ST00990*
      DO 105 I=1,I0                                                     ST01000*
      X=PHI(I,J,K)                                                      ST01010*
      IF(T(I,J,K).LE.0.) X=-99.                                         ST01020*
  105 EL(I,J,K)=X                                                       ST01030*
C                                                                       ST01040*
      IF (IHEAD.NE.ICHK(2) .OR. ITRHED.NE.0) GO TO 120                  ST01050*
C     ---PRINT HEAD TABLES---                                           ST01060*
      DO 110 K=1,K0                                                     ST01070
      IF (LEVEL2(K).EQ.0) GO TO 110                                     ST01080*
      WRITE (IPR,9004) K, KP, KT, SUM                                   ST01090*
      DO 110 I=1,I0                                                     ST01100
```

```
      WRITE (IPR,9005) I,(EL(I,J,K),J=1,J0)                            ST01110*
  110 CONTINUE                                                         ST01120*
C                                                                      ST01130*
  120 IF(IFINAL.NE.2) GO TO 160                                        ST01140*
C                                                                      ST01150*
C    --- SAVE HEAD ON DISK FILE (PLOT-FILE FORMAT)---                  ST01160*
      IF(ISVHED.NE.1) GO TO 191                                        ST01170*
        IDSK=15                                                        ST01180*
      DO 190 K=1,K0                                                    ST01190*
        WRITE(IDSK) SETID(1), SETID(2), SUM, K, I0, J0,                ST01200*
     1   ((EL(I,J,K), I=1,I0), J=1,J0)                                 ST01210*
  190 CONTINUE                                                         ST01220*
      WRITE(6,9002) IDSK, KP, KT, SUM                                  ST01230*
  191 CONTINUE                                                         ST01240*
C                                                                      ST01250*
C     ---SAVE HEAD AND MASS BALANCE (BOTH DOUBLE PRECISION) ON         ST01260*
C        UNFORMATTED DISK FILE.                                        ST01270*
      IF(IDK2.NE.ICHK(5)) GO TO 130                                    ST01280*
        IDSK=4                                                         ST01290*
      DO 196 K=1,K0                                                    ST01300*
  196 WRITE(IDSK) ((PHI(I,J,K), I=1,I0),J=1,J0)                        ST01310*
      WRITE(IDSK) SUM, SUMP, (((CFLX(I,J,K), I=1,9), J=1,2), K=1,K0)   ST01320*
      GO TO 151                                                        ST01330*
C                                                                      ST01340*
C     ---SAVE HEAD AND MASS BALANCE ON FORMATTED DISK FILE---          ST01350*
  130 IF (IPU2.NE.ICHK(9)) GO TO 160                                   ST01360
        IDSK=3                                                         ST01370*
      WRITE (IDSK,230) SUM,SUMP                                        ST01380*
      DO 132 K=1,K0                                                    ST01390*
      DO 132 J=1,2                                                     ST01400*
  132 WRITE(IDSK,230) (CFLX(I,J,K), I=1,9)                             ST01410*
      DO 150 K=1,K0                                                    ST01420
      DO 150 I=1,I0                                                    ST01430*
  150 WRITE (IDSK,220)  (PHI(I,J,K),J=1,J0)                            ST01440*
C                                                                      ST01450*
  151 WRITE(6,9001) IDSK, KP, KT, SUM                                  ST01460*
C                                                                      ST01470*
  160 CONTINUE                                                         ST01480*
C                                                                      ST01490*
  998 RETURN                                                           ST01500*
C                                                                      ST01510
C                                                                      ST01520
  220 FORMAT (8F10.4)                                                  ST01530*
  230 FORMAT (1P,4E20.13)                                              ST01540*
 9001 FORMAT(  '-*** HEAD AND MASS BALANCE VALUES WRITTEN ON FILE NO. ',ST01550*
     1 I2,  ' FOR STRESS PERIOD = ',I4,  '  TIME STEP = ',I4, / 5X,    ST01560*
     2   '  SIMULATION TIME = ',F14.2,  ' ***')                        ST01570*
 9002 FORMAT(  '-*** HEAD VALUES WRITTEN ON FILE NO. ',I2,  ' (SINGLE-PRST01580*
     1ECISION PLOT-FILE FORMAT) FOR STRESS PERIOD ',I4,  ', TIME STEP ' ST01590*
     2, I4 / 5X, '  SIMULATION TIME = ', F14.2,  ' ***')               ST01600*
 9004 FORMAT (1H1,20X, 24HCOMPUTED HEAD FOR LAYER ,I2,                 ST01610*
     1 19H.  STRESS PERIOD = ,I4, 14H, TIME STEP = ,I4, 20H, SIMULATION ST01620*
     2TIME = ,F14.2 / 21X, 101(1H-) / 5H0 ROW )                        ST01630*
 9005 FORMAT(1H0,I4, 12F10.2 / (5X, 12F10.2))                          ST01640*
 9006 FORMAT(1H1,20X, 19HDRAWDOWN FOR LAYER ,I2, 19H.  STRESS PERIOD = ,ST01650*
```

```
1 I4, 14H, TIME STEP = ,I4, 20H, SIMULATION TIME = ,F14.2 /          ST01660*
2 21X, 96(1H-) / 5H0 ROW )                                           ST01670*
 END                                                                 ST01680
```

```
      SUBROUTINE VSEC (IOPT, SETNAM, PHI, V, T)                          VS00010*
C...PRINT VALUES FOR THE MODEL PARAMETER PHI ALONG VERTICAL              VS00020*
C   SECTIONS THRU A 3-D GW FLOW MODEL.                                   VS00030*
C                                                                        VS00040*
      DIMENSION  PHI(1), V(1), T(1), LAYSEC(20,2), NSNOD(20),            VS00050*
     1     NODSEC(500,2), SETNAM(2), SETID(2)                            VS00060*
C                                                                        VS00070*
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,      NPER,KTH,ITMAX,LENGTH,KP,NVS00080*
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCVS00090*
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN   VS00100*
     3,NCD,IPFLX,NIJ,NNOD,LWTABL,NTR,LRECH,ISVHED,ISVHDX,ISVFLX,NVSEC   VS00110*
     4,NRIV,IA1,JA1,KA1,NDRAIN,IA2,JA2,KA2,ISALT,IA3,JA3,KA3,NTAD       VS00120*
     5,NTRIAL,ITRHED,ISTEDY,NSUBAL,IBALAY                               VS00130*
C                                                                        VS00140*
      COMMON /SPARAM/ TMAX, CDLT, DELT, ERR, TEST, SUM, SUMP,            VS00150*
     1  BETA1, WMAX1, ERMULT, STEADY, DELT1                              VS00160*
C                                                                        VS00170*
      DATA  MXSEC, MXNOD, INIT, MXNPAG /20, 500, 0, 11/                  VS00180*
     1,     SETID /4HVCOO,4HRD   /                                       VS00190*
     2,     IDSK /16/                                                    VS00200*
C                                                                        VS00210*
      IF(IOPT.GT.0) GO TO 100                                            VS00220*
C                                                                        VS00230*
      WRITE(6,9000) NVSEC                                                VS00240*
C                                                                        VS00250*
      IF(NVSEC.LE.0) GO TO 998                                           VS00260*
      IF(NVSEC.GT.MXSEC) GO TO 900                                       VS00270*
C                                                                        VS00280*
C...READ RECORDS 6 AND 7 IN GROUP I DATA:  COORDINATES OF VERTICAL       VS00290*
C   SECTIONS.                                                            VS00300*
        INIT=1                                                           VS00310*
        KK=0                                                             VS00320*
      DO 20 J=1,NVSEC                                                    VS00330*
        READ(5,2913) LAYSEC(J,1), LAYSEC(J,2), NSNOD(J)                  VS00340*
        WRITE(6,9001) J, LAYSEC(J,1), LAYSEC(J,2), NSNOD(J)              VS00350*
        JJ=KK+1                                                          VS00360*
        KK=KK+NSNOD(J)                                                   VS00370*
        IF(KK.GT.MXNOD) GO TO 901                                        VS00380*
        READ(5,2913) (NODSEC(K,1), NODSEC(K,2), K=JJ,KK)                 VS00390*
C                                                                        VS00400*
        L2=NSNOD(J)                                                      VS00410*
        NPAG=L2/MXNPAG                                                   VS00420*
        IF((NPAG*MXNPAG).LT.L2) NPAG=NPAG+1                              VS00430*
        L2=JJ-1                                                          VS00440*
C                                                                        VS00450*
        DO 15 NP=1,NPAG                                                  VS00460*
        L1=L2+1                                                          VS00470*
        L2=L2+MXNPAG                                                     VS00480*
        IF(L2.GT.KK) L2=KK                                               VS00490*
   15   WRITE(6,9002) (NODSEC(K,1), NODSEC(K,2), K=L1,L2)                VS00500*
   20 CONTINUE                                                           VS00510*
      GO TO 998                                                          VS00520*
C                                                                        VS00530*
C...PRINT VALUES FOR THE MODEL PARAMETER PHI ALONG VERTICAL SECTIONS.    VS00540*
  100 IF(INIT.EQ.0) GO TO 902                                            VS00550*
```

```
C
        LNOD=0
      DO 2860 NV=1,NVSEC
        JL1=LAYSEC(NV,1)
        JL2=LAYSEC(NV,2)
        L3=LNOD
        L2=NSNOD(NV)
        LNOD=LNOD+L2
        NPAG=L2/MXNPAG
        IF ((MXNPAG*NPAG).LT.L2) NPAG=NPAG+1
C
        L2=L3+1
        L4=0
        K=JL2+1
      DO 2850 L1=JL1,JL2
        K=K-1
        I3=(K-1)*NIJ
      DO 2850 JJ=L2,LNOD
        I=NODSEC(JJ,1)
        J=NODSEC(JJ,2)
        N=I+(J-1)*I0+I3
        L4=L4+1
 2850   V(L4)=PHI(N)
C
C...SAVE ON DISK FILE THE PARAMETER VALUES FOR EACH VERTICAL SECTION.
      IF (ISVHDX.GT.0) WRITE(IDSK) SETNAM, SUM, NV, JL1, JL2, NSNOD(NV),
     1    (V(JJ), JJ=1,L4)
C     ---NOTE THAT  L4=(JL2-JL1+1)*NSNOD(NV)---
C
      DO 2859 NP=1,NPAG
        L2=L3+1
        L3=L3+MXNPAG
        IF (L3.GT.LNOD) L3=LNOD
        WRITE (6,2915) NV, SETNAM, NP, (NODSEC(J,1), NODSEC(J,2),
     1     J=L2,L3)
        K=JL2+1
      DO 2858 L1=JL1,JL2
        K=K-1
        I3=(K-1)*NIJ
        L4=0
      DO 2857 JJ=L2,L3
        I=NODSEC(JJ,1)
        J=NODSEC(JJ,2)
        N=I+(J-1)*I0+I3
        L4=L4+1
        V(L4)=PHI(N)
 2857 CONTINUE
        WRITE (6,2916) K, (V(J), J=1,L4)
 2858 CONTINUE
 2859 CONTINUE
      IF(ISVHDX.GT.0) WRITE(6,3003) SETNAM, NV, IDSK, SUM
 2860 CONTINUE
C
      GO TO 998
C
```

```
C...ERROR MESSAGES.                                                     VS01110*
  900 WRITE(6,9900) MXSEC                                               VS01120*
      GO TO 999                                                         VS01130*
  901 WRITE(6,9901) MXNOD                                               VS01140*
      GO TO 999                                                         VS01150*
  902 WRITE(6,9902)                                                     VS01160*
      GO TO 999                                                         VS01170*
C                                                                       VS01180*
  998 RETURN                                                            VS01190*
  999 STOP                                                              VS01200*
C                                                                       VS01210*
 2913 FORMAT(26I3)                                                      VS01220*
 2915 FORMAT(1H1,23X, 34HVALUES ALONG VERTICAL SECTION NO. ,I2,         VS01230*
     1  18H  FOR PARAMETER = , 2A4, 36X,                                VS01240*
     1 8HPAGE NO.,I3 / 24X, 62(1H-), 36X, 11(1H-) /                     VS01250*
     2 1H0,6X, 21(1H.), 77HPARAMETER VALUES FOR SPECIFIED ROW AND COLUMNVS01260*
     3 COORDINATES (I,J) IN EACH LAYER, 22(1H.) / 1X, 5HLAYER,          VS01270*
     4 11(2X, 1H(, I3, 1H,, I3, 1H) ) /)                                VS01280*
 2916 FORMAT(1X, I3, 2X, 11(1X, F10.4))                                 VS01290*
 3003 FORMAT(  '-*** VALUES OF PARAMETER = ',2A4,  ', FOR VERTICAL SECTIVS01300*
     1ON = ',I2,  ', ARE SAVED ON FORTRAN FILE NO. ',I2,                VS01310*
     2  '.  SIMULATION TIME = ',F14.2, ' ***' )                         VS01320*
 9000 FORMAT( '-DATA FROM RECORDS 6 AND 7 IN GROUP I DATA: VERTICAL SECTVS01330*
     1IONS FOR WHICH MODEL PARAMETERS ARE PRINTED'/ 1X, 99(1H-)/        VS01340*
     2 1H0,4X,  'TOTAL NO. SECTIONS = ',I3)                             VS01350*
C9000 FORMAT(  '-*** SPECIFICATIONS FOR PRINTING VALUES OF A MODEL PARAMVS01360*
C    1ETER ALONG ',I3,   ' VERTICAL SECTIONS ***')                      VS01370*
 9001 FORMAT(1H-,4X, 'SECTION NO. = ',I3,   '    BOTTOM LAYER = ',I3,   VS01380*
     1 '    TOP LAYER = ',I3,   '    NO. CELLS ALONG SECTION = ',I4 /   VS01390*
     2 1H0,4X, 28(1H.), 63HROW AND COLUMN COORDINATES (I,J) FOR EACH LAYVS01400*
     3ER OF THIS SECTION, 28(1H.) /  5X,    11(9H( I , J ), 2X) )       VS01410*
 9002 FORMAT(2X, 11(I7, 1H,, I3) )                                      VS01420*
 9900 FORMAT( '-***ERROR IN SUBR. VSEC*** NO. VERTICAL SECTIONS EXCEEDS VS01430*
     1', I3)                                                            VS01440*
 9901 FORMAT( '-***ERROR IN SUBR. VSEC*** TOTAL OF CELLS FOR ALL SECTIONVS01450*
     1S EXCEEDS ', I5)                                                  VS01460*
 9902 FORMAT( '-***ERROR IN SUBR. VSEC*** INCORRECT VALUE SPECIFIED FOR VS01470*
     1IOPT.')                                                           VS01480*
      END                                                               VS01490*
```

```
      SUBROUTINE ZFWSW (ZI, ZOLD, ZTOP, ZBOT, ZDAMP, RF, SPHI,          ZF00010*
     1   THKMIN, THKPCT, THKRST, THKFUL)                                ZF00020*
C...COMPUTE DEPTH TO THE FRESHWATER-SALTWATER INTERFACE.                ZF00030*
C                                                                       ZF00040*
C...THIS ROUTINE IS CALLED FROM SUBROUTINE FWSW.                        ZF00050*
C                                                                       ZF00060*
C    ZDAMP  = 0 FOR NO DAMPING WHEN COMPUTING INTERFACE POSITION.       ZF00070*
C           =<0 TO USE SINGLE DAMPING WHEN COMPUTING INTERFACE POSITION.ZF00080*
C           =>0 TO USE DOUBLE DAMPING WHEN COMPUTING INTERFACE POSITION.ZF00090*
C    NOTE: THE DAMPING FACTOR, ZDAMP, IS USED ONLY WHEN ZOLD LIES       ZF00100*
C          BETWEEN THE TOP  AND BOTTOM OF A CELL.                       ZF00110*
C                                                                       ZF00120*
      ZI=SPHI*RF                                                        ZF00130*
      IF(ZDAMP.EQ.0.) GO TO 180                                         ZF00140*
      IF((ZOLD.LE.ZTOP .AND. ZI.LE.ZTOP) .OR. (ZOLD.GE.ZBOT .AND. ZI.GE.ZF00150*
     1   ZBOT)) GO TO 180                                               ZF00160*
      IF((ZOLD.LE.ZTOP .AND. ZI.GE.ZTOP) .OR. (ZOLD.GT.ZBOT .AND. ZI.LE.ZF00170*
     1   ZBOT)) GO TO 171                                               ZF00180*
C                                                                       ZF00190*
      ZZ=ZI-ZOLD                                                        ZF00200*
      IF(ZDAMP.LT.0.) GO TO 160                                         ZF00210*
C                                                                       ZF00220*
      IF(ZOLD.GT.ZI) ZZ=-AMIN1(ZOLD-ZI, ZOLD-ZTOP)                      ZF00230*
      IF(ZOLD.LE.ZI) ZZ= AMIN1(ZI-ZOLD, ZBOT-ZOLD)                      ZF00240*
C                                                                       ZF00250*
C   ---COMPUTE DEPTH TO INTERFACE---                                    ZF00260*
  160 ZI=ZZ*ABS(ZDAMP) + ZOLD                                           ZF00270*
C                                                                       ZF00280*
C   ---ADJUST INTERFACE POSITION WHEN ZI BETWEEN ZTOP AND ZBOT---       ZF00290*
  170 IF(ZI.LE.ZTOP .OR. ZI.GE.ZBOT) GO TO 180                          ZF00300*
  171 THKF=ZI-ZTOP                                                      ZF00310*
      THK1=ZBOT-ZTOP                                                    ZF00320*
      IF(ZOLD.GT.ZTOP) GO TO 172                                        ZF00330*
        THKF=AMIN1(THK1*THKRST, THKF)                                   ZF00340*
        ZI=ZTOP+THKF                                                    ZF00350*
        GO TO 173                                                       ZF00360*
  172 IF(ZOLD.LE.ZBOT) GO TO 173                                        ZF00370*
        ZI=ZBOT                                                         ZF00380*
        GO TO 180                                                       ZF00390*
  173 IF(THKF.LE.AMAX1(THK1*THKPCT, THKMIN)) ZI=ZTOP                    ZF00400*
      IF(THKF.GE.THK1*THKFUL) ZI=ZBOT                                   ZF00410*
C                                                                       ZF00420*
  180 RETURN                                                            ZF00430*
      END                                                               ZF00440*
```