MODIFICATIONS AND CORRECTIONS TO THE FINITE-DIFFERENCE MODEL FOR SIMULATION OF THREE-DIMENSIONAL GROUND-WATER FLOW

By Lynn J. Torak

U.S. GEOLOGICAL SURVEY
Water-Resources Investigations 82-4025

1982

UNITED STATES DEPARTMENT OF THE INTERIOR
JAMES G. WATT, Secretary

GEOLOGICAL SURVEY
Dallas L. Peck, Director

For additional information write to:

U.S. Geological Survey
National Center, Mail Stop 411
12201 Sunrise Valley Drive
Reston, Virginia 22092

Copies of this report
can be purchased from:

Open-File Services Section
Western Distribution Branch
U.S. Geological Survey
Box 25046, Federal Center
Lakewood, Colorado 80225
(Telephone: (303) 234-5888)

MODIFICATIONS AND CORRECTIONS TO THE FINITE-DIFFERENCE MODEL FOR SIMULATION OF THREE-DIMENSIONAL GROUND-WATER FLOW

CONTENTS

ILLUSTRATIONS

ABSTRACT

This report describes modifications that are incorporated into the finite-difference model for simulation of ground-water flow in three dimensions by Trescott, and Trescott and Larson. These modifications extend the application of this model to simulations involving head-dependent sources and sinks (i.e., rivers, evapotranspiration, and springs or drains). Other modifications are made that enhance the iterative-solution process of the Strongly Implicit Procedure (SIP). An acceleration (or dampening) factor is introduced to the matrix equation that is to be solved, and optional methods of computing iteration parameters are incorporated into the original model.

This report also describes corrections to the model that eliminate errors in the equation formulation and in mass-balance computations for certain types of simulations. Additional data-input instructions, definitions of new variables, and a list of program-statement changes are given in the Appendix.

MODIFICATIONS AND CORRECTIONS TO THE FINITE-DIFFERENCE MODEL FOR SIMULATION OF THREE-DIMENSIONAL GROUND-WATER FLOW

INTRODUCTION

Users of the finite-difference model for simulation of three-dimensional ground-water flow by Trescott (1975), and Trescott and Larson (1976) have indicated a need for simulating head-dependent source-sink functions (i.e., rivers, evapotranspiration, and drains or springs). They have also exposed some shortcomings of the model which may cause errors in the equation formulation for certain types of problems. This report describes modifications that are incorporated into the standard 3-D model for simulating these types of functions and other modifications which, in general, may enhance convergence of an iterative solution by the Strongly Implicit Procedure (SIP). This report also contains corrections to the model that eliminate errors in (a) formulating coefficients for SIP when simulating a water-table aquifer with the Basic 3-D approach (Equation 3, Trescott, 1975); (b) formulating coefficients in the vertical direction when the Quasi 3-D approach is used (Equation 4, Trescott, 1975, with TK values input to the model); (c) computation of vertical flow; and (d) punched output format of the PHI matrix.

Two versions of the 3-D model are described in this report. One version contains all of the modifications and corrections listed above. The other version contains only corrections to the standard 3-D model as described in Trescott (1975), and Trescott and Larson (1976).

MODIFICATIONS

Head-Dependent Source-Sink Functions

(Notes pertaining to these modifications were originally distributed at the training course "Advanced Modeling of Ground-Water Flow," January 8-17, 1980, by Steven P. Larson.)

General mathematical form. The types of head-dependent source-sink functions that can be simulated with the modified finite-difference model are streams, evapotranspiration, and springs or drains. A general mathematical expression that can be used to approximate all of these functions is

$$QSS = CSS(HSS-h) \quad [L/T] \text{ or } [T^{-1}], \tag{1}$$

where QSS is the flow rate derived from the head-dependent function. CSS is a constant that describes the linear relationship between the head difference and the flow rate. CSS functions like a leakance coefficient similar to the TK values used in Trescott (1975) to simulate steady leakage from a confining layer. Specific definitions of CSS will be given later for each type of head-dependent source-sink function. HSS is the fixed head [L] that controls the flow rate, such as the water level in a stream or river, the depth below land where the evapotranspiration rate is zero, or the spring or drain discharge elevation. The variable, h, is the computed head [L] at a node in the aquifer model.

The dimensions assigned to CSS and to equation 1 are determined by the type of 3-D modeling approach that is taken. For the Modified or Quasi 3-D approach (Equation 4, page 4 of Trescott, 1975), each term in the partial differential equation for ground-water flow, including QSS, has the dimensions

[L/T], representing a volumetric flow rate per unit surface area. CSS has the dimensions $[T^{-1}]$ for this approach.

For the Basic 3-D approach (Equation 3, page 3, Trescott, 1975), each term represents a volumetric flow rate per unit volume. The head-dependent source-sink term, QSS, would appear on the right-hand side of this equation and would also have the dimensions $[T^{-1}]$. For this approach, CSS has the dimensions $[L^{-1}T^{-1}]$.

The general form of the head-dependent source-sink functions given by (1) is modified according to the type of function to be simulated; rivers, springs or drains, or evapotranspiration. Formulation in the 3-D finite-difference model of the general form of the head-dependent source-sink functions is described, and the extension of this formulation to the specific types of functions is discussed. The model has been modified so that only one formulation of these functions is needed in the code. Checking procedures in the model allow this formulation to simulate the specific types of head-dependent source-sink functions.

The variable names used in this discussion are identical to those used in the modified model, with the exception that PHI in the model is replaced by h in this discussion. The head-dependent source-sink functions have been formulated in the model so that their effects can be simulated in any layer.

Formulation in the 3-D model. The head-dependent source-sink function is incorporated into the finite-difference equation for node i,j,k (Equation 9, page 9 of Trescott, 1975) as

$$Bh_{i-1} + Dh_{j-1} + Eh + Fh_{j+1} + Hh_{i+1} + Zh_{k-1} + Sh_{k+1} = Q - QSS, \qquad (2)$$

where the subscripts not containing a "+1" or a "-1" are eliminated. Substituting (1) into (2) yields

$$Bh_{i-1} + Dh_{j-1} + Eh + Fh_{j+1} + Hh_{i+1} + Zh_{k-1} + Sh_{k+1} = Q - CSS(HSS-h). \quad (3)$$

The implicit part of the head-dependent function, CSS•h, is combined with Eh on the left-hand side of (3), and the explicit part, CSS•HSS, remains on the right-hand side, yielding

$$Bh_{i-1} + Dh_{j-1} + E'h + Fh_{j+1} + Hh_{i+1} + Zh_{k-1} + Sh_{k+1} = Q - QS, \quad (4)$$

where

$$E' = E - CSS \text{ and} \quad (5a)$$

$$QS = CSS \cdot HSS. \quad (5b)$$

Following the description of the numerical solution by SIP on pages 11-15 of Trescott (1975), equation 4 can be represented in matrix notation for all i,j,k as

$$\bar{\bar{A}}' \cdot \bar{h} = \bar{Q}'.$$

The matrix A' contains the coefficients B, D, E', F, H, S, and Z, and the vector Q' contains the additional term QS for the head-dependent function described by (5). From Equation 12 on page 12 of Trescott (1975), the iterative scheme,

$$(\bar{\bar{A}}' + \bar{\bar{B}})\bar{h}^n = \bar{Q}' + \bar{\bar{B}}\bar{h}^{n-1},$$

is placed in residual form by adding and subtracting $\bar{\bar{A}}'\bar{h}^{n-1}$ to the right-hand side:

$$(\bar{\bar{A}} + \bar{\bar{B}})\bar{\xi}^{n} = \bar{R}^{n-1}$$

where $\bar{\xi}^{n} = \bar{h}^{n} - \bar{h}^{n-1}$;

$$\bar{R}'^{n-1} = \bar{Q}' - \bar{\bar{A}}'\bar{h}^{n-1}.$$

The superscript n is an iteration index and $(\bar{\bar{B}})$ is a modifying matrix used in SIP so that $(\bar{\bar{A}} + \bar{\bar{B}})$ can be factored into a form that can be solved easily. The vector R'^{n-1} is computed in the model as the right-hand-side variable, RES, and contains the implicit and explicit parts of the head-dependent source-sink function given by (5). At each node where a head-dependent source-sink function is to be simulated, the modified model reformulates the E coefficient so that it contains the leakance coefficient CSS. The explicit part of the head-dependent function is also computed in the model and subtracted from the right-hand-side variable RES. This general formulation is modified according to the specific type of head-dependent source-sink function that is simulated.

Streams or rivers. The general form of the head-dependent source-sink function given by (1) is modified to simulate the interaction of a stream with an aquifer,

$$QSS = \begin{cases} CSS(HSS-h), & h > HSL & \text{(6a)} \\ CSS(HSS-HSL), & h \leq HSL. & \text{(6b)} \end{cases}$$

HSS is the elevation of the water level in the stream. HSL is a threshold elevation which limits the maximum flow rate to the aquifer, equation 6b, whenever the computed head is below this elevation. HSL is frequently chosen as the elevation of the bottom of the streambed, or as the value of h which corresponds to a hydraulic gradient of unity between the streambed and the aquifer. The relationship of HSS and HSL to the aquifer head is shown in figure 1.

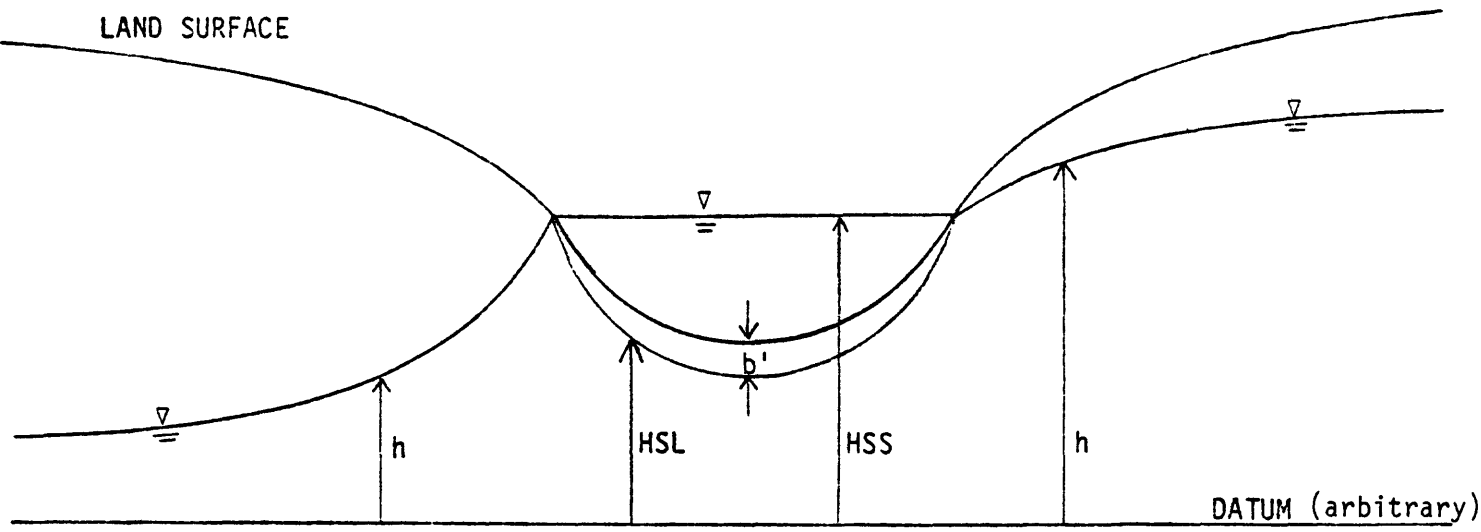

Figure 1.--Relationship between threshold elevations for head-dependent source-sink functions and aquifer head.

Another threshold elevation, HB, is embodied in the program modifications for head-dependent source-sink terms. This value, if used, would limit the maximum flow rate out of the aquifer and into the stream. However, as no physical basis exists for this type of effect, HB should be set equal to an arbitrarily high value so this threshold condition is eliminated. HB remains in the program logic for generality, so that only one formulation of head-dependent source-sink functions occurs in the code. The function of HB is described in a later section pertaining to evapotranspiration.

The streambed leakance coefficient, CSS, accounts for the percentage of the grid block covered by the streambed (A_s/A, see figure 2), the weighted harmonic mean of hydraulic conductivity between the stream bottom and the aquifer node (k'), and the vertical interval between the stream bottom and the node (b').

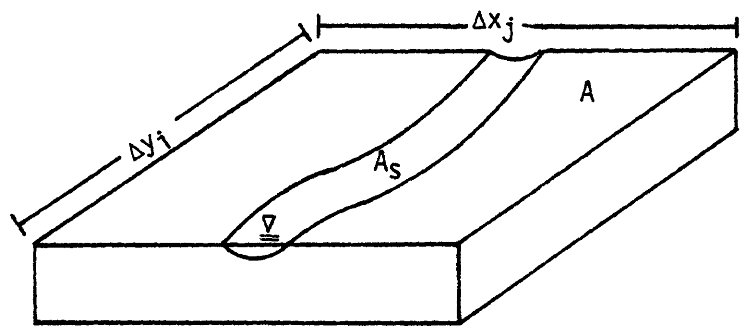

Figure 2.--Areal relationship between the streambed and the grid block used to compute CSS in equation 7.

For many problems k' is taken as the hydraulic conductivity of the streambed sediments and b' is taken as their thickness. All of these factors are lumped together in one term for each stream node as

$$CSS = \frac{k'}{b'} \cdot \frac{A_s}{A}. \tag{7}$$

A_s is the area of the streambed contained in the grid block and A is the area of the grid block. The expression of CSS in (7) is for the Modified or Quasi 3-D approach, Equation 4 of Trescott (1975). For the Basic 3-D approach, Equation 3 of Trescott (1975), CSS is divided by Δz_k, the thickness of the finite-difference layer where the steam node is located.

Formulation of the threshold condition of (6b) is accomplished by two steps in the model. First, the implicit and explicit parts of the general form of the head-dependent function are computed using equations 5. Second, a check is made to determine if the computed head from the previous iteration is lower than HSL. If the computed head is lower than HSL, the excess amount, CSS(HSL-h), is subtracted from QS, restricting the flow rate to the value given by (6b).

For h < HSL,

$$QS = CSS \cdot HSS - CSS(HSL-h), \tag{8}$$

Equation 4 becomes

$$Bh_{i-1} + Dh_{j-1} + Eh - CSSh + Fh_{j+1} + Hh_{i+1} + Zh_{k-1} + Sh_{k+1}$$
$$= Q - CSS \cdot HSS + CSS(HSL-h). \tag{9}$$

In the model, equation 9 is placed in residual form; that is, all terms of (9) are placed on the right side, and the left side is designated as the variable, RES,

$$\mathrm{RES} = -Bh_{i-1}^{n-1} - Dh_{j-1}^{n-1} - Eh^{n-1} + CSSh^{n-1} - Fh_{j+1}^{n-1} - Hh_{i+1}^{n-1} - Zh_{k-1}^{n-1} - Sh_{k+1}^{n-1}$$

$$+ Q - CSS \cdot HSS + CSS \cdot HSL - CSS \cdot h^{n-1}, \qquad (10)$$

where n-1 is the index for the iteration level. The goal of the iterative process is to bring the variable RES sufficiently close to zero so that (9) is essentially satisfied at all nodes of the mesh. The implicit parts of the head-dependent function in equation 10 (the $CSS \cdot h^{n-1}$ terms) cancel, limiting the maximum flow rate into the aquifer from the stream to the value CSS(HSS-HSL) in equation 6b.

For conditions where the aquifer head is above the threshold elevation, HSL, equation 6a is used, and the implicit and explicit parts of the head-dependent function are formulated by equations 5.

My approach to these modifications has been to formulate the implicit part of the head-dependent function, E', in equation 5a, at every time step and iteration for the appropriate nodes, even though a threshold condition may exist which causes the formulation to be totally explicit. If the computed heads cause a threshold condition to exist, then corrections are made to the residual RES. The implicit formulation of the head-dependent term remains on the E coefficient; thus, the left-hand-side matrix (A' + B) is not changed during the simulation. An alternate approach would be to only place the implicit part of the head-dependent function on the E coefficient when the computed head is between the threshold elevations. If the computed head is close to either threshold, its value may cross the threshold elevation repeatedly during the iterative-solution process. Thus, on successive iterations, the E coefficient and the (A' + B) matrix may or may not contain the implicit part of the head-dependent function. This approach may cause numerical oscillations about a solution as the computed heads approach the threshold elevations. These oscillations tend to be reduced by formulating E' all the time at the nodes where this function is to be simulated.

This approach yields a matrix (A' + B) which is not entirely correct when the threshold condition is invoked. As the formulation should be totally explicit, no CSS term should appear on the main diagonal. The implicit term is compensated for in the residual, but not in the left-hand-side matrix. However, convergence does occur with this formulation in simulations that did not converge using the alternate approach described above. A comparison between computed results obtained from this formulation and those obtained from the alternate approach when convergence occurred shows insignificant differences in values of head and volumetric flow rates.

Evapotranspiration. The general form of the head-dependent source-sink function given in (1) is modified to simulate evapotranspiration,

$$QSS = \begin{cases} CSS(HSS-HB), & h \geq HB \quad (11a) \\ CSS(HSS-h), & HSS < h < HB \quad (11b) \\ 0. , & h \leq HSS, \quad (11c) \end{cases}$$

where HB is the land-surface elevation and HSS is a water-table elevation below which evapotranspiration from the water table is zero. The constant, CSS, is computed as q_{ET}^{max}/d_{ET} $[T^{-1}]$, for the Modified or Quasi 3-D approach, Equation 4 of Trescott (1975). For the Basic 3-D approach, Equation 3 of Trescott (1975), CSS is divided by Δz_k, the thickness of the finite-difference layer where ET is simulated. Here q_{ET}^{max} is the maximum evapotranspiration rate per unit area (such as 84 inches/year), [L/T] and d_{ET} [L] is the depth of the water table below land surface where the ET rate is zero; that is, d_{ET} = HB-HSS (see figure 3).

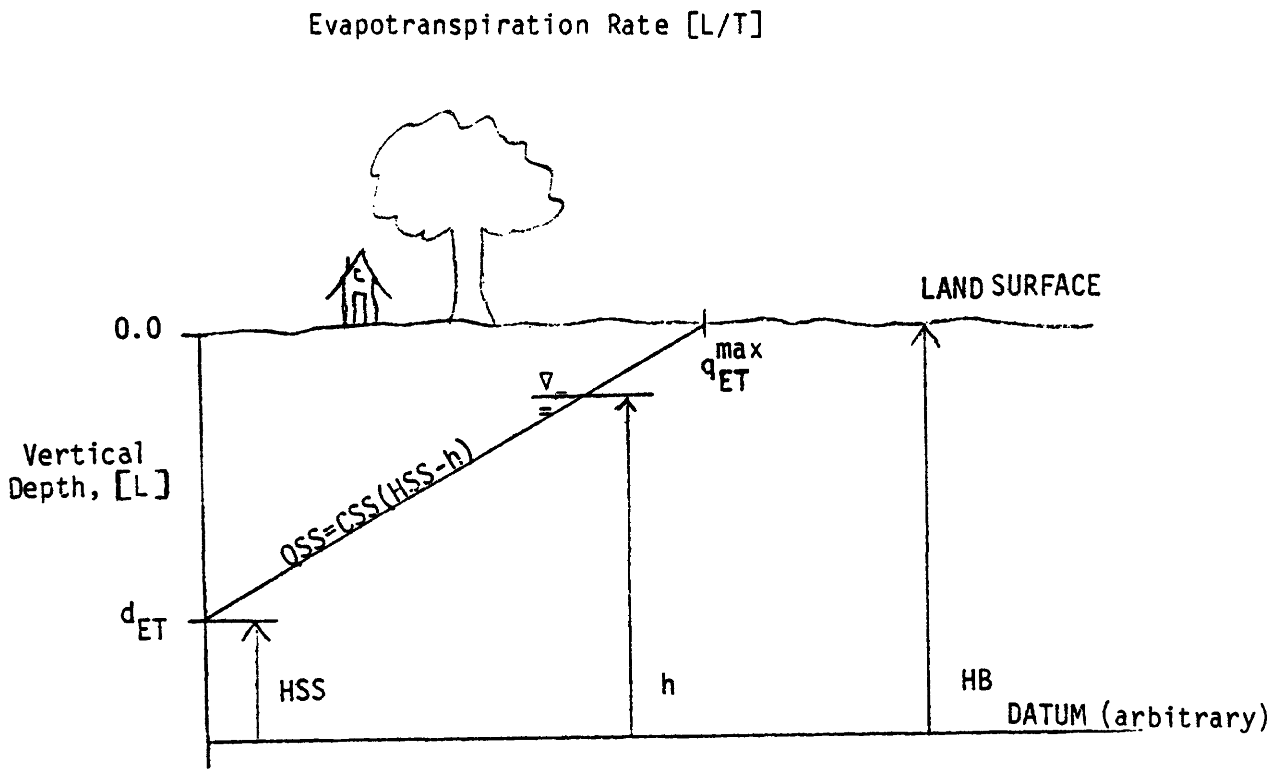

Figure 3.--Relationship of threshold elevations and aquifer head to the evapotranspiration rate, QSS.

This formulation assumes the ET rate is a linear function of the head difference, (HSS-h), when the computed head is between land surface and HSS. This relationship is given in figure 3 and in (11b) as the expression for QSS.

A threshold condition for the ET rate is met when the computed head is above land surface, HB. For this condition, the ET rate is limited to q_{ET}^{max}, computed by (11a). When the aquifer head falls below HSS, the ET rate will be zero, as given by (11c).

The formulation of evapotranspiration in the model is identical to the formulation for streams and rivers. The implicit and explicit parts of the general form of the head-dependent source-sink function are computed using (5) for all elevations of head at the nodes where ET is simulated. The upper threshold condition, $h > HB$, is formulated in the model in the same fashion as for streams and rivers; the excessive amount of the ET rate, given by CSS(HB-h), is subtracted from QS as in (8), limiting the maximum ET rate to CSS(HSS-HB). The variable, QS, is then subtracted from the residual, RES, yielding equation 10.

Formulation of the lower threshold condition, $h < HSS$, is accomplished in the model by specifying HSL = HSS during data input. The model checks if h is less than HSL. If so, a correction is applied to QS which was formulated as in (5b) for the general case,

$$QS = QS - CSS(HSS - h), \tag{12}$$

causing the ET rate to be zero.

When the residual is computed, all head-dependent source-sink terms cancel, and an expression equivalent to (11c) results,

$$RES = -Bh_{i-1}^{n-1} - Dh_{j-1}^{n-1} - Eh^{n-1} + CSSh^{n-1} - Fh_{j+1}^{n-1} - Hh_{i+1}^{n-1} - Zh_{k-1}^{n-1} - Sh_{k+1}^{n-1}$$
$$+ Q - CSS \cdot HSS + CSS \cdot HSS - CSS \cdot h^{n-1}.$$

Springs or drains. The form of the head-dependent source-sink function used for springs or drains is identical to the expression of QSS given by (11b-c) for evapotranspiration.

A spring (or drain) leakance coefficient, CSS, is used to describe the linear relationship assumed to exist between the head difference, (HSS-h), and the spring (or drain) discharge rate. This linear relationship is assumed to be for water levels greater than HSS. HSS is defined as the spring or drain discharge elevation, below which the discharge from the spring or drain is zero. For springs, as in streams, the physical basis for limiting the maximum discharge with the threshold HB may not exist, but the formulation remains in the model for generality. Therefore, a value of HB may be set arbitrarily high so the spring discharge rate is not limited.

Threshold conditions for spring or drain discharge are formulated in the model using the same convention as for evapotranspiration. The lower threshold, HSL, is not used for spring or drain discharge but its value is input as the value of HSS to allow the model to make the appropriate correction to QS given in (12).

Features. The input requirements for the head-dependent functions are designed so that a unique value of the leakance coefficient, CSS, can be input at each node where these functions are simulated. One variable, CSS, is used to represent either a streambed leakance, evapotranspiration rate, or a spring or drain leakance coefficient. Because of this design, only one type of head-dependent function can be simulated at any node. Values of CSS are input to the model in the same manner as the array data in Group III, page III-5, of Trescott (1975), either with a single parameter card per layer, or with a parameter card followed by a matrix of CSS values. If the Basic 3-D approach is used (Equation 3 of Trescott, 1975), then CSS must be divided by Δz_k, the thickness of the finite-difference layer where the head-dependent function is located. Thus, the dimensions of CSS change with respect to the type of modeling approach that is taken. (The additional data-input instructions for these modifications are given in Appendix I). Values of CSS are output either in matrix form or as one value per layer, depending upon the input convention selected.

The threshold elevations, HB, HSS, and HSL, are input and stored only for those nodes where a head-dependent function is to be simulated. This eliminates the allocation of storage for 3 three-dimensional arrays. The nodes simulating head-dependent functions are indexed by positive CSS values. Thus, the CSS array also serves as an indicator array for the head-dependent source-sink functions. The order in which threshold values are input to the model follows the indexing scheme for the finite-difference mesh with the normal ordering of nodes given on page 5 of the documentation by Trescott (1975). Threshold elevations are printed out for each node, for verification, according to the same node-ordering scheme used as input.

The arrays and vectors used to store the head-dependent terms in the model are incorporated into the general storage vector, Y, and allocation of storage for these terms occurs at the time of execution. A description of the Y vector is given on page II-1 of Trescott (1975), and the Y locations for the new arrays and vectors are given in Appendix II.

Volumetric flow rates derived from head-dependent sources and sinks are printed out for each node with the standard printout of mass balance. The total volumetric flow rate and the total volume of water derived from head-dependent sources and sinks are also printed out with separate output for sources and sinks. In addition, the cumulative volume of water discharged by evapotranspiration and springs during the simulation, and the combined ET and spring-discharge rate for the last time step, is printed out in the location provided for ET by the original 3-D model.

Discussion. Checkout simulations with head-dependent source-sink functions in all layers were conducted using a steady-state model. The flow rates computed by the model were nearly identical to those obtained by hand calculations.

It would be impossible to test these modifications against every hydrogeologic system for which the effects of head-dependent source-sink functions are to be considered. Therefore, the user must exercise care in applying them to field problems to assure the specific type of function properly simulates the observed effects.

For the simulations I have conducted, I found that the inclusion of head-dependent terms into the matrix equation of (3) tended to slow down the iterative process of obtaining a solution using SIP. That is, more iterations were required per time step to achieve convergence. Because of this, an acceleration factor has been incorporated into the computation of the residual, RES, to enhance convergence. This factor is discussed in the next section. Also, other methods of computing the maximum iteration parameter have been added to the model in an effort to speed the convergence process. These methods are described in a later section.

Acceleration or Dampening Factor for the Strongly Implicit Procedure (SIP)

(This modification had previously been added to the standard 3-D model by Steven P. Larson for use in the training course "Advanced Modeling of Ground-Water Flow," and for numerical experiments, see Larson and Trescott, 1977.)

To reduce the number of iterations needed for SIP to converge when head-dependent source-sink functions are simulated, an acceleration factor, HMAX, is introduced into the calculation of the residual, RES. From Equation 13 on page 13 of Trescott (1975), the residual form of the finite-difference matrix equation is stated as

$$(\bar{\bar{A}}' + \bar{\bar{B}})\, \bar{\xi}^n = \bar{R}'^{n-1},$$

where $\bar{R}'^{n-1}$ is the right-hand-side variable, RES, previously discussed, and $\bar{\xi}^n$ is the head change during the n^{th} iteration. The factor HMAX multiplies the right-hand side of this equation, changing the magnitude of the ξ^n values which are obtained by solution using SIP. Overestimating the residual, RES, by multiplying it by a real number larger than 1.0 will, in most cases, increase the increments of head change that are solved for on each iteration of SIP, thus speeding up the convergence process. In this sense, HMAX is used as an acceleration factor to reduce the number of iterations needed for convergence.

For simulations I have conducted using an acceleration factor and the head-dependent functions, a value of HMAX = 1.2 seemed to give better convergence than simulations using smaller or larger values. However, the optimum value of HMAX to use is undoubtedly problem dependent. The value of HMAX can be increased (or decreased) slightly from 1.0 on successive simulations while noting the number of iterations per time step required for solution. The number of iterations should decrease for those simulations, and then begin to increase, indicating that the optimum value of HMAX had been bypassed.

HMAX can also be used as an acceleration factor for simulations that do not invoke the head-dependent option. Larson and Trescott (1977) note that for two-dimensional problems, a value of HMAX greater than 1 may expedite convergence for simulations involving strongly anisotropic conditions. For water-table conditions, HMAX may be used as a dampening factor, its value less than unity, so that the iterative solution is approached slowly, reducing the incidence of nodes dropping out of the solution due to oscillations in SIP.

Other descriptions of an acceleration or dampening factor, β', for SIP in the two-dimensional case can be found in Trescott, Pinder, and Larson (1976, p. 26-29) and in Larson and Trescott (1977). This factor is identical to HMAX used here in the 3-D model.

Optional Methods for Computing the Maximum Iteration Parameter

For some aquifer simulations, other methods of computing the maximum iteration parameter, WMAX, gave faster convergence than the method presently used in the 3-D model of Trescott (1975). Several approaches to computing the maximum iteration parameter are recognized in the literature and can be applied to nonhomogeneous, anisotropic aquifers discretized with unequal grid spacings. A brief description of these approaches is presented here to provide a basis for the methods incorporated into the 3-D model. It should be noted, however, that there is no theoretically based method which determines exactly the 'best' value for WMAX. All methods described here were empirically derived, with the exception of the formula by Peaceman (1977), which was developed for the Alternating Direction Implicit Procedure (ADIP).

In a paper describing SIP for two-dimensional equations, Stone (1968) states that the best maximum value for ω, WMAX, can be computed from

$$(1 - \omega_{max}) = \underset{\text{over grid}}{\text{Min}} \left[\frac{2\Delta x^2}{1 + \frac{K_y \Delta x^2}{K_x \Delta y^2}} , \frac{2\Delta y^2}{1 + \frac{K_x \Delta y^2}{K_y \Delta x^2}} \right], \qquad (13)$$

where

Δx, Δy are the grid spacings, and

K_x, K_y are the hydraulic conductivities in the x and y directions, respectively.

The assumptions made by Stone in arriving at equation 13 are that K_x and K_y and Δx and Δy are constant over the model area.

The equation for computing the maximum iteration parameter developed by Stone (1968) is "similar" to the one given below by Peaceman (1977) for ADIP.

$$(1 - \omega_{max}) = \underset{\text{over grid}}{\text{Min}} \quad \frac{\Pi^2}{2J^2(1+\frac{AY}{AX})}, \quad \frac{\Pi^2}{2I^2(1+\frac{AX}{AY})} , \qquad (14)$$

where

J is the number of grid spacings in the X direction

I is the number of grid spacings in the Y direction

AY are the coefficients B and H = $K_y/\Delta y^2$

AX are the coefficients D and F = $K_x/\Delta x^2$, K_x and K_y are hydraulic conductivities in the x and y directions, Δx and Δy are grid spacings

$(1 + \frac{AY}{AX})$ and $(1 + \frac{AX}{AY})$ are the same as the demoninators in (13).

These equations were developed for problems involving two-dimensional grids, not three, and although (14) was derived for ADIP, it has been proven effective in the solution by SIP (see Weinstein, Stone, and Kwan, 1969).

In cases where K_x, K_y, and Δx, Δy vary over the grid, Stone (1968) found it sufficient to compute local values of ω_{max} at each grid point by equation 13 and to use the arithmetic average of these values for the value of ω_{max}. Cooley (1974) found that values of ω_{max} computed by this procedure resulted in convergence of the SIP solution scheme in cases where divergence occurred using the absolute minimum over the grid for $(1-\omega_{max})$.

The method of computing an average ω_{max} for two-dimensional problems has been extended to the three-dimensional case and is incorporated into the model as a modification. The equivalent of equation 14 for three dimensions is given by Weinstein, Stone, and Kwan (1969), as

$$(1-\omega_{max}) = \underset{\text{over grid}}{\text{Min}} \left[\frac{\Pi^2}{2J^2(1+\rho_1)}, \frac{\Pi^2}{2I^2(1+\rho_2)}, \frac{\Pi^2}{2K^2(1+\rho_3)} \right], \qquad (15)$$

where K is the number of layers, and I and J are the same as for equation 14.

The equations for the ρ terms are

$$\rho_1 = \frac{K_{yy}(i,j,k)(\Delta x_j)^2}{K_{xx}(i,j,k)(\Delta y_i)^2} + \frac{K_{zz}(i,j,k)(\Delta x_j)^2}{K_{xx}(i,j,k)(\Delta z_k)^2}$$

$$\rho_2 = \frac{K_{xx}(i,j,k)(\Delta y_i)^2}{K_{yy}(i,j,k)(\Delta x_j)^2} + \frac{K_{zz}(i,j,k)(\Delta y_i)^2}{K_{yy}(i,j,k)(\Delta z_k)^2}$$

$$\rho_3 = \frac{K_{xx(i,j,k)}(\Delta z_k)^2}{K_{zz(i,j,k)}(\Delta x_j)^2} + \frac{K_{yy(i,j,k)}(\Delta z_k)^2}{K_{zz(i,j,k)}(\Delta y_i)^2}.$$

The first term in the expressions for ρ_1 and ρ_2 is identical to the second term in the denominators of (13), and to $\frac{AY}{AX}$ and $\frac{AX}{AY}$ in (14). By extending the 2-D expression for ω_{max} to three dimensions, Weinstein, Stone, and Kwan (1969) introduced a third term to (13) that considers the ratio of hydraulic conductivity and grid spacing between the x-z and y-z planes when computing the value of ω_{max}. Also, another term appeared in the denominators of (13) and (14) yielding the full expressions for the ρ terms. These are the second terms in the expressions for ρ_1 and ρ_2 which account for the ratio of hydraulic conductivity and grid spacing between the z-x and z-y planes.

The assumptions made by Stone (1968) which allowed Weinstein, Stone, and Kwan (1969) to develop equation 15 for computing ω_{max} were that $K_x = K_y = K_z =$ constant and $\Delta x = \Delta y = \Delta z =$ constant over the grid (see p. 285 of Weinstein, Stone, and Kwan, 1969). Therefore, only the values of I, J, or K, the number of grid spacings in the x, y, or z directions, determine which of the three terms in (15) would be used to compute ω_{max}. For cases where values of the ρ terms vary over the grid, either because of nonhomogeneity or variable grid spacing, Weinstein, Stone, and Kwan (1969) suggest computing local values of ω_{max} at each active grid point using (15); then selecting the maximum value as ω_{max}.

In the original 3-D model, the expressions for the ρ terms are changed so that the coefficients D, F, B, H, Z, and S are used to compute ω_{max}. These coefficients are defined by Equations 8, on page 8 of the documentation by Trescott (1975). This change allows harmonic means of transmissivities (Modified or Quasi 3-D approach) or hydraulic conductivities (Basic 3-D approach) to be used in computing the ρ terms instead of the nodal values of these parameters.

The original 3-D model of Trescott (1975) changes the computation of ρ_1, ρ_2, and ρ_3 further by using only the first term in their expressions (see the expressions for the ρ terms in equations 16). These truncated ρ terms are then used in (15) to compute ω_{max} at each active grid point, and the maximum value over the grid is selected as ω_{max}. Modifications to the model allow ρ_1, ρ_2, and ρ_3 to be computed using the full formulation (both terms for each ρ) given by equations 16. The new formulation computes the ρ terms as

$$\rho_1 = \frac{\text{Max }[B,H]}{\text{Min }[D,F]} + \frac{\text{Max }[S,Z]}{\text{Min }[D,F]} \tag{16a}$$

$$\rho_2 = \frac{\text{Max }[D,F]}{\text{Min }[B,H]} + \frac{\text{Max }[S,Z]}{\text{Min }[B,H]} \tag{16b}$$

$$\rho_3 = \frac{\text{Max }[D,F]}{\text{Min }[S,Z]} + \frac{\text{Max }[B,H]}{\text{Min }[S,Z]}. \tag{16c}$$

In the modified model, several options are available for computing ω_{max}. Its value may be calculated as the maximum value over the entire grid of active nodes, or as the arithmetic average of the local maxima for ω_{max}, as previously described. Both of these methods for determining ω_{max} use the full formulation of the ρ terms, as defined by equations 16. The original method of computing ω_{max} from the truncated expressions of the ρ terms in (16) is also available.

My experience with these methods of computing ω_{max} (WMAX) indicates the arithmetic-average approach may give faster convergence than the original method for aquifer problems consisting of widely varying transmissivities (or hydraulic conductivities) in combination with uneven grid spacings. An average ω_{max} would eliminate the possibility of one node with eccentric aquifer and grid-spacing properties from dominating the computation of ω_{max}, thereby generating a sequence of iteration parameters which may be inappropriate (give slow convergence) to the solution by SIP. For problems simulating isotropic layers in conjunction with grid spacings in the x, y, and z directions of the same order of magnitude, the computation of ω_{max} as the maximum value over the grid with the full formulation of the ρ terms in (16), may give faster convergence than the original method. Some guidelines for using the approach taken by Trescott in computing ω_{max} are given in his documentation (see page 25 of Trescott, 1975).

For those modelers who wish to bypass all computations of WMAX, the model was modified to allow its value to be input by the user. A sequence of iteration parameters is then generated from this value, as it is for all values of WMAX, using Equation 24 on page 25 of Trescott (1975). However, an error exists in this equation, and the correct expression is

$$W_{\ell+1} = 1 - (1-W_{max})^{\ell/(L-1)}, \ \ell = 0,1,\ldots,L-1.$$

If the value of WMAX is being input by the user, the following hints from Weinstein, Stone, and Kwan (1969) may be helpful. If the sequence of iteration parameters computed by (15) causes divergence, then the value $(1-\omega_{max})$ should be multiplied by a factor of 2 to 10. If the iteration is slow to converge, $(1-\omega_{max})$ should be divided by a factor of 2 to 10.

CORRECTIONS

Dimensional Nonhomogeneity of Coefficients in ENTRY ITER

In this section of the program, the coefficients D, F, B, H, Z, and SU are calculated so that the maximum iteration parameter (WMAX) can be computed. The original code does not check for the particular equation being solved, either Equation 4, for the Modified or Quasi 3-D approach, or Equation 3, for the Basic 3-D approach. Only the Basic 3-D approach requires the coefficients in the vertical direction, Z and SU, to be divided by the thickness of the layer, DELZ(K); however, this division occurs for both Equation 3 and Equation 4 formulations.

To correct this problem, a check is made for the type of equation being solved in the model before Z and SU are divided by DELZ(K). If the Equation 4 approach is taken, division of these coefficients is not performed. A similar check was incorporated into the original 3-D model (see page XIX of Trescott and Larson, 1976) where these coefficients are computed again for the solution by SIP. The uncorrected version of ENTRY ITER would usually not cause any errors in the calculation of WMAX. As previously discussed, the vertical components in the formulation of ρ_1 and ρ_2 had been eliminated by Trescott in the original model, and ρ_3 usually does not determine the value for WMAX. However, if the complete formulation of the ρ terms given by equations 16 are used, the improper formulation of Z and SU would always cause errors in determining the maximum iteration parameter.

Use of Transmissivity for Computing T-Coefficients for Water-Table Simulations Using Equation 3

The 3-D model was originally designed to solve Equation 4 on page 4 of Trescott (1975). This approach requires transmissivity to be input to the model. For water-table simulations, transmissivity values are updated after each iteration based on the newly computed thickness of the top layer. The 3-D model can also be used to solve Equation 3 on page 3 of Trescott (1975). With this formulation, hydraulic conductivities are input; thus transmissivities are not required. Regardless of whether Equation 3 or Equation 4 is used in a water-table simulation, the program calculates transmissivities and T-coefficients (TR, TC, and TK) for the upper layer as the initial start-up conditions and as updates after each iteration. These values are then used to compute the coefficients D, F, B, H, Z, and SU which are input to SIP. Because of this error, transmissivities are used to form TR and TC coefficients in the top layer of a water-table problem with Equation 3, when the correct formulation requires hydraulic conductivities. The TR and TC coefficients for the top layer of a water-table aquifer simulated with Equation 3 are calculated incorrectly as weighted harmonic means of transmissivities instead of the correct formulation using hydraulic conductivities.

The TK coefficent is also in error in the original model for water-table simulations using Equation 3. This term is defined as the weighted harmonic mean of vertical hydraulic conductivity and layer thickness; however, the transmissivity of the top layer is used in the model.

This condition has been corrected by converting the updated (or initial) values of transmissivity to hydraulic conductivities in ENTRY TRANS, where the transmissivity calculations occur. The computation of transmissivity is necessary to determine if a node has become inactive by dewatering; therefore, ENTRY TRANS is still entered on each iteration for both Equation 3 and Equation 4 approaches.

As part of the above correction, the computation of the coefficients TR, TC, and TK needs only to be performed once for water-table simulations involving Equation 3. Therefore, the model has been changed so that portion of the subroutine which computes these values, ENTRY TCOF, is only passed through once under these conditions.

Despite these changes which allow a correct formulation of coefficients to be made for water-table simulations using Equation 3, this approach to a water-table problem should be avoided. The flow equation solved with the Equation 3 approach does not take into account the changing saturated thickness of the top layer. Because transmissivities are not used in this approach, values for the coefficients TR, TC, and TK do not change as the thickness of the water-table layer changes. These coefficients only change when the node goes dry; then they are set to zero.

Occurrence of a Nonzero TK Value Between Active and Inactive Nodes

When the Quasi 3-D approach to aquifer simulation is used, and one value of TK on a parameter card is assigned to the entire matrix of TK's, an error in the coefficient formulation for SIP will result if there is an inactive node (zero transmissivity) above or below an active node. A nonzero value for TK will exist between an active and an inactive node representing a vertical component of flow. This causes an error in the formulation of the coefficients, SU and Z, as these coefficients should be set to zero. There should be no contribution of flow to a node from either above or below when the nodes above or below that node are inactive.

Several checks have been inserted in subroutine SOLVE where these coefficients are computed so that, if the transmissivity of the node directly above (or below) a particular node is zero, the SU (or Z) coefficient will be assigned a value of zero.

This correction is not necessary if the Modified 3-D approach is taken. The program computes values of TK and will automatically set TK = 0 if one of the vertical hydraulic conductivities is zero. This causes either the Z or SU coefficient to be zero.

Computation of Vertical Flow from Inactive or Constant Head Nodes

In ENTRY CWRITE of the CHECKI subroutine, a computation of vertical flow is made at all "interior" nodes in the two top layers and in the two bottom layers. (An interior node is one that is not located on the border of inactive nodes placed around the model.) No checks are made to determine if these "interior" nodes are inactive (zero transmissivity), or if they are constant-head nodes. Errors in the volumetric balance will result if vertical flow is computed between active and inactive nodes (if TK is nonzero), or between two constant-head nodes. Checks have been incorporated into the computation of vertical flow eliminating the problem.

Assignment of N3=1 in COEF Subroutine

Statement COF 195 which set N3=1 has been eliminated. In its place, the value of N3 is passed to ENTRY TCOF of subroutine COEF as a parameter on the CALL statement in the MAIN program (statement MAN1700). Thus, N3 is set equal to 1 on the first pass through ENTRY TRANS and ENTRY TCOF when these parts of the COEF subroutine are called from the MAIN program. A value of zero for N3 is passed to ENTRY TRANS on subsequent CALL's from subroutine SOLVE. This correction supersedes that described as item II in the memorandum of May 6, 1977.

The change indicated by that memorandum appeared to be a source of confusion concerning the value and function of N3 during the simulation. The change indicated here causes N3 to be consistently given a value as it is needed in ENTRY TRANS and ENTRY TCOF.

Incompatibility Between Formats for Punched Output of PHI Matrix and Data-Input Requirements

The punched output of heads generated in ENTRY OUTPUT of subroutine STEP is not compatible with the format used to read these cards as input to the PHI matrix for continuation of a previous simulation. The output format statement STP1350 should be changed to FORMAT (8F10.4) for compatibility.

Improper Dimensioning of JFLO and FLOW

The dimensions of the array JFLO has been changed to (3,NCD) and the dimensions of the vector FLOW has been changed to (NCD) in statement CHK 120 and CHK 130 of subroutine CHECKI in both the corrected and modified models. The original model dimensions these terms with NCH, the number of constant-head nodes, instead of NCD which is defined as MAXO(1,NCH) in statement MAN0510. An error may occur in dimensioning JFLO and FLOW with NCH, as its value may be zero if no constant-head nodes are to be simulated.

In the modified model, statement MAN0510 has also been changed so that NCD is the maximum value of 1, NCH, and NHD. The number of nodes where head-dependent sources or sinks are simulated, NHD, has been added to the maximum-value function in this statement. JFLO and FLOW, now dimensioned with NCD, store the nodal location and the volumetric flow rates of the head-dependent source-sink nodes, respectively, in addition to their original use with constant-head nodes. This change also requires NCD to be included in the COMMON/INTEGR/ statement of all subroutines for both the corrected and modified models.

Program statements constituting modifications or corrections to the standard 3-D model are identified by an asterisk (*) or a digit (0-9) in column 80. Every effort was made to preserve the original program sequence numbers in columns 73-79, so that the modified and/or corrected version can be compared with the standard model.

Many of the corrections, or the need for them, have been brought to my attention by users of the 3-D model. I would appreciate hearing from anyone implementing these modifications and/or corrections into the model, and am interested in learning about the types of field problems for which the modified model is applied.

REFERENCES

Cooley, R. L., 1974, Finite-element solutions for the equations of ground-water flow: Technical Report Series H-W, Hydrology and Water Resources Publication No. 18, Center for Water Resources Research, Desert Research Institute, University of Nevada System, 134 p.

Larson, S. P., and Trescott, P. C., 1977, Solution of water-table and anisotropic flow problems by using the strongly implicit procedure: Journal of Research, U.S. Geological Survey, Vol. 5, No. 6, Nov-Dec, 1977, p. 815-821.

Peaceman, D. W., 1977, Fundamentals of numerical reservoir simulation: Elsevier Scientific Publishing Company, 176 p.

Stone, H. L., 1968, Iterative solution of implicit approximations of multidimensional partial differential equations: Soc. Indus. Applied Math., Jour. Numer. Anal., Vol. 5, No. 3, p. 530-558.

Trescott, P. C., 1975, Documentation of finite-difference model for simulation of three-dimensional ground-water flow: U.S. Geological Survey Open-File Report 75-438, 32 p.

Trescott, P. C., and Larson, S. P., 1976, Documentation of finite-difference model for simulation of three-dimensional ground-water flow, supplement to Open-File Report 75-438: 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, Chap. C1, 116 p.

Weinstein, H. C., Stone, H. L., and Kwan, T. V., 1969, Iterative procedure for solution of systems of parabolic and elliptic equations in three dimensions: Industrial Engineering Chemistry Fundamentals, Vol. 8, No. 2, p. 281-287.

APPENDIX I

Additional Data-Input Instructions for Modifications (The sequence of Group and Card numbers identifying the additional data inputs is compatable with the data deck instructions given in Appendix III of Trescott, 1975.)

Group I: Title, Simulation Options, and Problem Dimensions

Card	Columns	Format	Variable	Definition
3	61-70	I10	NHD	Number of nodes where head-dependent source-sink terms are computed.
4	61-64	A4	IHDEP	Code HDEP for head-dependent source-sink option.

Group II: Scalar Parameters

Card	Columns	Format	Variable	Definition
6	1-20	G20.10	HDVPT	Cumulative volume of head-dependent sources. For continuation of a previous simulation using the head-dependent option, replace this card with the fourth card of punched output from the previous run. See description of cards 3, 4, and 5, p. III-4 of Trescott (1975). For a new simulation, if HDEP is specified, insert a blank card. Not needed when HDEP is not specified.

Group III: Array Data

Card	Columns	Format	Variable	Definition
8a	1-80	20F4.0	CSS(I,J,K)	Leakance values that describe the relationship between the hydraulic gradient and the head-dependent flux. For evapotranspiration (Equation 4, Trescott, 1975), code the maximum ET rate/depth where the ET rate is zero, $[T^{-1}]$. See discussion for other leakances. Include a parameter card with this data set.
8b	1-30	3G10.0	HB(N) HSS(N) HSL(N)	Values for the upper, middle, and lower threshold heads for head-dependent source-sink terms. See discussion for

Card	Columns	Format	Variable	Definition
8b (Cont'd)				special data inputs of these values for simulating ET, rivers, and springs or drains. For ET and springs or drains, set HSL = HSS. For streams and springs, HB is set to an arbitrarily high value so this threshold is not invoked. Do not include a parameter card with this data set. Number of cards for this data set is the value of NHD. The order of these inputs follows the normal ordering of nodes given on page 5 of Trescott (1975). N = 1, NHD.
12	1-20	2F10.0	WMAX	WMAX is the value of the maximum iteration parameter ($0 < WMAX < 1$), generally a value slightly less than 1 used to speed convergence. If 1. is coded, WMAX is an indicator that causes iteration parameters to be computed according to the original formulation. If 2. is coded, iteration parameters are computed according to equations 15 and 16. A value of 3. causes WMAX to be computed as the average WMAX over the entire grid, using the full formulation of ρ terms given by (16).
			HMAX	HMAX is the acceleration (or dampening) factor used in SIP to reduce oscillations and speed convergence (see explanation of HMAX and β' in Trescott, Pinder, and Larson, 1976). Generally, $0 < HMAX < 2$.

APPENDIX II

Definition of New Variables Used In The Modified Model

Variable	Definition
CSS	A constant describing the linear relationship between the hydraulic gradient and the head-dependent flux, $[T^{-1}]$ or $[L^{-1}T^{-1}]$ depending on whether Equation 3 or Equation 4 of Trescott (1975) is used. Its location in the Y vector is L(26).
HB	Value of head for the upper threshold of a head-dependent flux [L]. Its location in the Y vector is L(28).
HDD	Hydraulic head used to compute HDRT.
HDEP	Value of IHDEP to be coded for head-dependent option.
HDFN	Total volumetric flow rate derived from head-dependent sinks $[L^3/T]$.
HDFP	Total volumetric flow rate derived from head-dependent sources $[L^3/T]$.
HDRT	Nodal volumetric flow rate for a head-dependent source or sink $[L^3/T]$. Its location in the Y vector is L(30).
HDVNT	Cumulative volume of water derived from all head-dependent sinks $[L^3]$.
HDVPT	Cumulative volume of water derived from all head-dependent sources $[L^3]$.
HMAX	Acceleration or dampening factor for RES.
HSL	Value of head for the lower threshold of a head-dependent flux [L]. Its location in the Y vector is L(29).
HSS	Value of head for the middle threshold of a head-dependent flux [L]. Its location in the Y vector is L(27).
ICHK(13)	Element of vector that contains the value HDEP, used to invoke head-dependent option.
IE,JE,KE	Dimensions of CSS array.
IHDEP	Option for formulating head-dependent fluxes.
IR	Index for locating nodes that contain head-dependent sinks for printout of mass balance.

Variable	Definition
IRR	Counter for IR.
IWM	Indicator for computing a new WMIN.
LHD	Vector for locating a node containing a head-dependent source or sink in the SIP reverse algorithm. Its location in the Y vector is L(31).
NHD	Number of nodes where head-dependent fluxes are to be computed.
NNHD	Counter for NHD.
NWM	Counter for WMIN (integer variable).
QS	Head-dependent flux [L/T].
SUMWM	Sum of WMIN.
WMNUM	Counter for WMIN (real variable).

APPENDIX III

LIST OF MODEL CORRECTIONS

PROGRAM STATEMENTS CONSTITUTING CORRECTIONS TO THE ORIGINAL MODEL OF THREE-DIMENSIONAL GROUND-WATER FLOW OF TRESCOTT (1975), AND TRESCOTT AND LARSON (1976), ARE IDENTIFIED BY AN ASTERISK (*) OR A DIGIT (0-9) IN COLUMN 80. A FEW UNCORRECTED STATEMENTS ARE INCLUDED IN THIS LIST TO ASSURE PROPER SEQUENCING OF THE CORRECTIONS.

***** MAIN *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NMAN0150
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCMAN0160
     2H,IDK1,IDK2,IWATER,IORE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  MAN0170*
     3NCD                                                               MAN0171*
C     ---COMPUTE T COEFFICIENTS---                                      MAN1690
      CALL TCOF(1)                                                      MAN1700*
C                                                                       MAN1710
```

***** DATAI *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NDAT0160
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCDAT0170
     2H,IDK1,IDK2,IWATER,IORE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  DAT0180*
     3NCD                                                               DAT0181*
```

***** STEP *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NSTP 160
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCSTP 170
     2H,IDK1,IDK2,IWATER,IORE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  STP 180*
     3NCD                                                               STP 181*
  220 FORMAT (8F10.4)                                                   STP1350*
```

***** SOLVE *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NSP3 160
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCSP3 170
     2H,IDK1,IDK2,IWATER,IORE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  SP3 180*
     3NCD                                                               SP3 181*
      SU=0.D0                                                           SP3 480
      Z=0.D0                                                            SP3 490
      IF (K.EQ.1) GO TO 5                                               SP3 491*
      IF (T(N-NIJ).EQ.0.) GO TO 5                                       SP3 492*
      Z=TK(N-NIJ)                                                       SP3 493*
      IF (IEQN.EQ.ICHK(11)) Z=Z/DELZ(K)                                 SP3 494*
    5 IF (K.EQ.K0) GO TO 10                                             SP3 495*
      IF (T(N+NIJ).EQ.0.) GO TO 10                                      SP3 496*
      SU=TK(N)                                                          SP3 497*
      IF (IEQN.EQ.ICHK(11)) SU=SU/DELZ(K)                               SP3 498*
   10 CONTINUE                                                          SP3 560
      IF(K.EQ.1) GO TO 124                                              SP31361
      IF(T(NKB).EQ.0.) GO TO 124                                        SP313611
      Z=TK(NKB)                                                         SP31362
```

```
      IF(IEQN.EQ.ICHK(11)) Z=Z/DELZ(K)                                  SP31363
  124 IF(K.EQ.K0) GO TO 125                                             SP31371
      IF(T(NKA).EQ.0.) GO TO 125                                        SP313711
      SU=TK(N)                                                          SP31372
      IF(K.EQ.1) GO TO 174                                              SP32231
      IF(T(NKB).EQ.0.) GO TO 174                                        SP322311
      Z=TK(NKB)                                                         SP32232
  174 IF(K.EQ.K0) GO TO 175                                             SP32241
      IF(T(NKA).EQ.0.) GO TO 175                                        SP322411
      SU=TK(N)                                                          SP32242
```

***** COEF *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NCOF 150
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCCOF 160
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  COF 170
     3NCD                                                               COF 171*
      T(I,J,K0)=PERM(I,J)*(PHI(I,J,K0)-BOTTOM(I,J))                     COF 300
      IF (T(I,J,K0).GT.0.) GO TO 5                                      COF 310*
      PHI(I,J,K0)=1.D30                                                 COF 410
      GO TO 10                                                          COF 411*
    5 IF (IEQN.EQ.ICHK(11)) T(I,J,K0)=PERM(I,J)                         COF 412*
   10 CONTINUE                                                          COF 420
      IF (N3.EQ.1) RETURN                                               COF 430
      IF (IEQN.EQ.ICHK(11).AND.N3.EQ.0) RETURN                          COF 431*
      N1=K0                                                             COF 440
C     ********************                                              COF 490
      ENTRY TCOF(N3)                                                    COF 500*
C     ********************                                              COF 510
```

***** CHECKI *****

```
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(K0,3), JFLO(NCD,3), FLOCHK 120*
     3W(NCD), QRE(IQ,JQ)                                                CHK 130*
C                                                                       CHK 140
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NCHK 150
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCCHK 160
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  CHK 170*
     3NCD                                                               CHK 171*
      DO 250 J=2,J1                                                     CHK1490
      IF(T(I,J,1).EQ.0..OR.T(I,J,2).EQ.0.) GO TO 245                    CHK1491*
      IF(S(I,J,1).LT.0..AND.S(I,J,2).LT.0.) GO TO 245                   CHK1492*
      X=X+(PHI(I,J,1)-PHI(I,J,2))*TK(I,J,1)*DELX(J)*DELY(I)             CHK1500
  245 IF(T(I,J,K1).EQ.0..OR.T(I,J,K0).EQ.0.) GO TO 250                  CHK1501*
      IF(S(I,J,K1).LT.0..AND.S(I,J,K0).LT.0.) GO TO 250                 CHK1502*
      Y=Y+(PHI(I,J,K1)-PHI(I,J,K0))*TK(I,J,K1)*DELX(J)*DELY(I)          CHK1520
  250 CONTINUE                                                          CHK1521*
```

***** PRINTAI *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NPRN 130
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCPRN 140
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,  PRN 150*
     3NCD                                                               PRN 151*
```

APPENDIX IV

LIST OF MODEL MODIFICATIONS

PROGRAM STATEMENTS CONSTITUTING MODIFICATIONS (AND CORRECTIONS) TO THE ORIGINAL MODEL OF THREE-DIMENSIONAL GROUND-WATER FLOW OF TRESCOTT (1975), AND TRESCOTT AND LARSON (1976), ARE IDENTIFIED BY AN ASTERISK (*) OR A DIGIT (0-9) IN COLUMN 80. A FEW UNMODIFIED STATEMENTS ARE INCLUDED IN THIS LIST TO ASSURE PROPER SEQUENCING OF THE MODIFICATIONS.

***** MAIN *****

```
      DIMENSION Y(102000), L(31), HEADNG(33), NAME(42), INFT(2,2), IOFT(MAN0100*
     19,4), DUM(3)                                                      MAN0110
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NMAN0150
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCMAN0160
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHMAN0170*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                          MAN0171*
      WRITE (6,190) HEADNG                                              MAN0360
      READ (5,160) I0,J0,K0,ITMAX,NCH,NHD                               MAN0370*
      WRITE (6,180) I0,J0,K0,ITMAX,NCH,NHD                              MAN0380*
      READ (5,210) IDRAW,IHEAD,IFLO,IDK1,IDK2,IWATER,IQRE,IPU1,IPU2,ITK MAN0390
     1,IEQN,ITL,IHDEP                                                   MAN0391*
      WRITE (6,220) IDRAW,IHEAD,IFLO,IDK1,IDK2,IWATER,IQRE,IPU1,IPU2,ITKMAN0400
     1,IEQN,ITL,IHDEP                                                   MAN0401*
      IERR=0                                                            MAN0410
      IMAX=MAX0(I0,J0)                                                  MAN0500
      NCD=MAX0(1,NCH,NHD)                                               MAN0510*
      JQ=1                                                              MAN1110
   70 IF(IHDEP.NE.ICHK(13)) GO TO 75                                    MAN1111*
      L(26)=ISUM                                                        MAN1112*
      ISUM=ISUM+ISIZ                                                    MAN1113*
      L(27)=ISUM                                                        MAN1114*
      ISUM=ISUM+NHD                                                     MAN1115*
      L(28)=ISUM                                                        MAN1116*
      ISUM=ISUM+NHD                                                     MAN1117*
      L(29)=ISUM                                                        MAN11171
      ISUM=ISUM+NHD                                                     MAN11172
      L(30)=ISUM                                                        MAN11173
      ISUM=ISUM+NHD                                                     MAN11174
      L(31)=ISUM                                                        MAN11175
      ISUM=ISUM+ISIZ                                                    MAN11176
      IE=I0                                                             MAN1118*
      JE=J0                                                             MAN1119*
      KE=K0                                                             MAN1120*
      GO TO 77                                                          MAN1121*
   75 L(26)=ISUM                                                        MAN1122*
      L(27)=ISUM                                                        MAN1123*
      L(28)=ISUM                                                        MAN1124*
      L(29)=ISUM                                                        MAN1125*
      L(30)=ISUM                                                        MAN1126*
      L(31)=ISUM                                                        MAN1127*
      IE=1                                                              MAN1128*
      JE=1                                                              MAN1129*
      KE=1                                                              MAN1130*
```

```
      NHD=1                                                             MAN1131*
   77 WRITE (6,170) ISUM                                                MAN1132*
C                                                                       MAN1133*
C     ---PASS INITIAL ADDRESSES OF ARRAYS TO SUBROUTINES---             MAN1140
      CALL DATAI(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))MAN1150
     1,Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(23)),Y(L(MAN1160
     224)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)))                MAN1170*
      CALL STEP(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)),MAN1180
     1Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(18)),Y(L(2MAN1190
     20)))                                                              MAN1200
      CALL SOLVE(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7))MAN1210
     1,Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(10)),Y(L(MAN1220
     211)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(20)),Y(L(25)),Y(L(26)),Y(L(27)MAN1230*
     3),Y(L(28)),Y(L(29)),Y(L(31)))                                     MAN1231*
      CALL COEF(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)),MAN1240
     1Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(23)),Y(L(2MAN1250
     24)),Y(L(25)))                                                     MAN1260
      CALL CHECKI(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)MAN1270
     1),Y(L(8)),Y(L(9)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(19)),Y(L(21)),Y(LMAN1280
     2(22)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(30)))      MAN1290*
      CALL PRNTAI(Y(L(1)),Y(L(2)),Y(L(4)),Y(L(5)),Y(L(9)),Y(L(15)),Y(L(1MAN1300
     16)))                                                              MAN1310
     137),IRN,DUM)                                                      MAN1630
C                                                                       MAN16301
C     ---READ LEAKANCES FOR HEAD-DEPENDENT TERMS---                     MAN16302
      IF (IHDEP.NE.ICHK(13)) GO TO 138                                  MAN16303
      DO 132 K=1,KO                                                     MAN16304
      LOC=L(26)+(K-1)*NIJ                                               MAN16305
  132 CALL ARRAY  (Y(LOC),INFT(1,1),IOFT(1,3),24HHEAD-DEPENDENT LEAKANCEMAN16306
     *S,IRN,DUM)                                                        MAN16307
C                                                                       MAN16308
C      ---READ THRESHOLD VALUES FOR HEAD-DEPENDENT SOURCE-SINK TERMS--- MAN16309
      CALL HDPDAT                                                       MAN16310
C                                                                       MAN16311
  138 CALL MDAT                                                         MAN1640*
C                                                                       MAN1650
C     ---COMPUTE TRANSMISSIVITY FOR UNCONFINED LAYER---                 MAN1660
      IF (IWATER.EQ.ICHK(6)) CALL TRANS(1)                              MAN1670
C                                                                       MAN1680
C     ---COMPUTE T COEFFICIENTS---                                      MAN1690
      CALL TCOF(1)                                                      MAN1700*
C                                                                       MAN1710
  160 FORMAT (5I10,10X,I10)                                             MAN2020*
  170 FORMAT ('0',54X,'WORDS OF VECTOR Y USED =',I7)                    MAN2030
  180 FORMAT ('0',62X,'NUMBER OF ROWS =',I5/60X,'NUMBER OF COLUMNS =',I5MAN2040
     1/61X,'NUMBER OF LAYERS =',I5//39X,'MAXIMUM PERMITTED NUMBER OF ITEMAN2050
     2RATIONS =',I5//48X,'NUMBER OF CONSTANT HEAD NODES =',I5,//,35X,'NUMAN2060*
     3MBER OF HEAD-DEPENDENT SOURCE-SINK NODES =',I5)                   MAN2061*
  190 FORMAT ('1',33A4)                                                 MAN2070
  200 FORMAT (20A4)                                                     MAN2080
  210 FORMAT (16(A4,1X))                                                MAN2090
  220 FORMAT ('-SIMULATION OPTIONS:  ',13(A4,4X))                       MAN2100*
  230 FORMAT (1H0,44X,'DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORSMAN2110
```

***** DATAI *****

```
      SUBROUTINE DATAI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACDAT0010
     1T,PERM,BOTTOM,QRE,CSS,HSS,HB,HSL)                                 DAT0020*
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(K0,3), PERM(IP,JP), BOTDAT0130
     3TOM(IP,JP), QRE(IQ,JQ), TF(3), A(I0,J0), IN(6), IOFT(9), INFT(2), DAT0140*
     4CSS(IE,JE,KE),HSS(NHD),HB(NHD),HSL(NHD)                           DAT0141*
```

```
C                                                                         DAT0150
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NDAT0160
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCDAT0170
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHDAT0180*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                            DAT0181*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR                 DAT0190
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)                        DAT0200
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT,   DAT0210*
     1HDVNT,HDVPT                                                         DAT0211*
      READ (5,450) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT0400
     1XT,FLXNT,HDVNT                                                      DAT0410*
      HDVPT=0.                                                            DAT0411*
      IF (IHDEP.EQ.ICHK(13)) READ (5,450) HDVPT                           DAT0412*
      IF (IDK1.EQ.ICHK(4)) GO TO 20                                       DAT0420
   20 READ (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT0520
     1XT,FLXNT,HDVNT,HDVPT                                                DAT0531*
      RETURN                                                              DAT0950
C     *********************                                               DAT09501
      ENTRY HDPDAT                                                        DAT09502
C     *********************                                               DAT09503
      WRITE (6,480)                                                       DAT09504
      N=0                                                                 DAT09505
      DO 142 K=1,K0                                                       DAT09506
      DO 142 I=2,I1                                                       DAT09507
      DO 142 J=2,J1                                                       DAT09508
      IF (CSS(I,J,K).EQ.0.) GO TO 142                                     DAT09509
      N=N+1                                                               DAT09510
      IF (N.GT.NHD) GO TO 144                                             DAT09511
      READ (5,490) HB(N),HSS(N),HSL(N)                                    DAT09512
      WRITE (6,500) K,I,J,HB(N),HSS(N),HSL(N)                             DAT09513
  142 CONTINUE                                                            DAT09514
      RETURN                                                              DAT09515
  144 WRITE (6,510) I,J,K                                                 DAT09516
      STOP                                                                DAT09517
C     *********************                                               DAT0960
      ENTRY MDAT                                                          DAT0970
C     *********************                                               DAT0980
     5I5.7,' PRINTED FOR LAYERS',9I2)                                     DAT2190
  480 FORMAT (1H0,40X,' THRESHOLD HEADS FOR HEAD-DEPENDENT SOURCE-SINK TDAT2191*
     1ERMS',//,36X,'LAYER',4X,'ROW',4X,'COLUMN',8X,'UPPER',7X,'MIDDLE',8DAT2192*
     2X,'LOWER',/,38X,'K',7X,'I',7X,'J',8X,'   (HB)    ',3X,'  (HSS)   ',DAT2193*
     34X,'  (HSL)    ',//)                                                DAT2194*
  490 FORMAT (3G10.0)                                                     DAT2195*
  500 FORMAT (1H ,33X,3(I5,3X),3(5X,F8.2))                                DAT2196*
  510 FORMAT (1H0,15X,'***** WARNING:  NUMBER OF THRESHOLD HEADS EXCEEDDAT2197*
     1S THE NUMBER OF NONZERO LEAKANCES (CSS) CODED *****',/,1H0,30X,'HEDAT2198*
     2AD-DEPENDENT TERMS STARTING WITH NODE',3I5,2X,'WERE NOT READ')     DAT2199*
      END                                                                 DAT2200-
```

***** STEP *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NSTP 160
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCSTP 170
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHSTP 180*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                            STP 181*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR                 STP 190
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)                        STP 200
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT,   STP 210*
     1HDVNT,HDVPT                                                         STP 211*
C     ---IF MAXIMUM ITERATIONS EXCEEDED,WRITE RESULTS ON DISK OR CARDS--STP 580
      IF (IDK2.EQ.ICHK(5)) WRITE (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHSTSTP 590
```

```
     1,CHDT,FLUXT,STORT,ETFLXT,FLXNT,HDVNT,HDVPT                        STP 600*
      IF (IPU2.NE.ICHK(9)) GO TO 20                                     STP 601*
      WRITE (7,230) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFSTP 610*
     1LXT,FLXNT,HDVNT,HDVPT                                             STP 620*
      IF (IDK2.EQ.ICHK(5)) WRITE (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHSTSTP1080
     1,CHDT,FLUXT,STORT,ETFLXT,FLXNT,HDVNT,HDVPT                        STP1090*
      WRITE (7,230) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFSTP1140
     1LXT,FLXNT,HDVNT,HDVPT                                             STP1150*
  220 FORMAT (8F10.4)                                                   STP1350*
```

***** SOLVE *****

```
      SUBROUTINE SOLVE(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACSP3  10
     1T,EL,FL,GL,V,XI,TEST3,QRE,CSS,HSS,HB,HSL,LHD)                     SP3  20*
      REAL *8PHI,RHO,B,D,F,H,Z,SU,RHOP,W,WMIN,RHO1,RHO2,RHO3,XPART,YPARTSP3  80
     1,ZPART,DMIN1,WMAX,XT,YT,ZT,DABS,DMAX1,DEN,TXM,TYM,TZM,SUMWM       SP3  90*
      REAL*8E,AL,BL,CL,A,C,G,WU,TU,U,DL,RES,SUPH,GLXI,ZPHI              SP3 100
C                                                                       SP3 110
      DIMENSION PHI(1), STRT(1), OLD(1), T(1), S(1), TR(1), TC(1), TK(1)SP3 120
     1, WELL(1), DELX(1), DELY(1), DELZ(1), FACT(K0,3), RHOP(20), TEST3(SP3 130
     21), EL(1), FL(1), GL(1), V(1), XI(1), QRE(1), CSS(1), HSS(1),HB(1)SP3 140*
     3,HSL(1),LHD(1),BOTTOM(1)                                          SP3 141*
C                                                                       SP3 150
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NSP3 160
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCSP3 170
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHSP3 180*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                          SP3 181*
      NIJ=I0*J0                                                         SP3 320
      READ (5,250) WMAX,HMAX                                            SP3 321*
      IF (WMAX.LT.1.) WRITE (6,260) WMAX,HMAX                           SP3 322*
      IF (WMAX.EQ.1.) WRITE (6,270) WMAX,HMAX                           SP3 323*
      IF (WMAX.EQ.2.) WRITE (6,280) WMAX,HMAX                           SP3 324*
      IF (WMAX.EQ.3.) WRITE (6,290) WMAX,HMAX                           SP3 325*
      IF (WMAX.LT.1.) GO TO 45                                          SP3 326*
      XT=3.141593**2/(2.*J2*J2)                                         SP3 330
      RHO3=0.D0                                                         SP3 380
      NWM=0                                                             SP3 381*
      SUMWM=0.D0                                                        SP3 382*
      DO 40 K=1,K0                                                      SP3 390
      DO 40 I=2,I1                                                      SP3 400
      DO 40 J=2,J1                                                      SP3 410
      IWM=0                                                             SP3 411*
      N=I+(J-1)*I0+(K-1)*NIJ                                            SP3 420
      Z=0.D0                                                            SP3 490
      IF (K.EQ.1) GO TO 5                                               SP3 491*
      IF(T(N-NIJ).EQ.0.) GO TO 5                                        SP3 492*
      Z=TK(N-NIJ)                                                       SP3 493*
      IF (IEQN.EQ.ICHK(11)) Z=Z/DELZ(K)                                 SP3 494*
    5 IF (K.EQ.K0) GO TO 10                                             SP3 495*
      IF (T(N+NIJ).EQ.0.) GO TO 10                                      SP3 496*
      SU=TK(N)                                                          SP3 497*
      IF (IEQN.EQ.ICHK(11)) SU=SU/DELZ(K)                               SP3 498*
   10 CONTINUE                                                          SP3 560
      TXM=DMAX1(D,F)                                                    SP3 570
      TYM=DMAX1(B,H)                                                    SP3 580
      TZM=DMAX1(SU,Z)                                                   SP3 590
      DEN=DMIN1(D,F)                                                    SP3 600
      IF (DEN.EQ.0.D0) DEN=TXM                                          SP3 610
      IF (DEN.EQ.0.D0) GO TO 20                                         SP3 620
      IF (WMAX.EQ.1.) RHO1=DMAX1(RHO1,TYM/DEN)                          SP3 630*
      IF (WMAX.EQ.2.) RHO1=DMAX1(RHO1,(TYM+TZM)/DEN)                    SP3 631*
```

```
      IF (WMAX.EQ.3.) RHO1=(TYM+TZM)/DEN                                 SP3 632*
      IWM=IWM+1                                                          SP3 633*
   20 DEN=DMIN1(B,H)                                                     SP3 640
      IF (DEN.EQ.0.D0) DEN=TYM                                           SP3 650
      IF (DEN.EQ.0.D0) GO TO 30                                          SP3 660
      IF (WMAX.EQ.1.) RHO2=DMAX1(RHO2,TXM/DEN)                           SP3 670*
      IF (WMAX.EQ.2.) RHO2=DMAX1(RHO2,(TXM+TZM)/DEN)                     SP3 671*
      IF (WMAX.EQ.3.) RHO2=(TXM+TZM)/DEN                                 SP3 672*
      IWM=IWM+1                                                          SP3 673*
   30 DEN=DMIN1(SU,Z)                                                    SP3 680
      IF (DEN.EQ.0.D0) DEN=TZM                                           SP3 690
      IF (DEN.EQ.0.D0) GO TO 39                                          SP3 700*
      IF (WMAX.EQ.1.) RHO3=DMAX1(RHO3,TXM/DEN)                           SP3 710*
      IF (WMAX.EQ.2.) RHO3=DMAX1(RHO3,(TXM+TYM)/DEN)                     SP3 711*
      IF (WMAX.EQ.3.) RHO3=(TXM+TYM)/DEN                                 SP3 712*
      IWM=IWM+1                                                          SP3 713*
   39 IF (WMAX.LT.3.) GO TO 40                                           SP3 714*
      IF (IWM.EQ.0) GO TO 40                                             SP3 715*
      XPART=XT/(1.D0+RHO1)                                               SP3 716*
      YPART=YT/(1.D0+RHO2)                                               SP3 717*
      ZPART=ZT/(1.D0+RHO3)                                               SP3 718*
      WMIN=DMIN1(XPART,YPART,ZPART)                                      SP3 719*
      SUMWM=SUMWM+WMIN                                                   SP3 720*
      NWM=NWM+1                                                          SP3 721*
   40 CONTINUE                                                           SP3 722*
      IF (WMAX.LT.3.) GO TO 42                                           SP3 723*
      WMNUM=NWM                                                          SP3 724*
      WMIN=SUMWM/WMNUM                                                   SP3 725*
      GO TO 44                                                           SP3 726*
   42 XPART=XT/(1.D0+RHO1)                                               SP3 727*
      YPART=YT/(1.D0+RHO2)                                               SP3 740
      ZPART=ZT/(1.D0+RHO3)                                               SP3 750
      WMIN=DMIN1(WMIN,XPART,YPART,ZPART)                                 SP3 760
   44 WMAX=1.D0-WMIN                                                     SP3 770*
   45 PJ=-1.                                                             SP3 780*
      DO 50 I=1,LENGTH                                                   SP3 790
      BIG=0.                                                             SP31000
      NNHD=0                                                             SP31001*
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 148                            SP31270*
      IF(K.EQ.1) GO TO 124                                               SP31361
      IF (T(NKB).EQ.0.) GO TO 124                                        SP313611
      Z=TK(NKB)                                                          SP31362
  124 IF(K.EQ.K0) GO TO 125                                              SP31371
      IF (T(NKA).EQ.0.) GO TO  125                                       SP313711
      SU=TK(N)                                                           SP31372
      QR=0.                                                              SP31390
      QS=0.                                                              SP31391*
  130 E=-B-D-F-H-SU-Z-RHO                                                SP31450
C     ---COMPUTE IMPLICIT AND EXPLICIT PARTS OF HEAD-DEPENDENT TERMS---  SP314501
      IF (IHDEP.NE.ICHK(13)) GO TO 138                                   SP314502
      LHD(N)=0                                                           SP314503
      IF (CSS(N).EQ.0.) GO TO 138                                        SP314504
      NNHD=NNHD+1                                                        SP314505
      IF (NNHD.GT.NHD) GO TO 138                                         SP314506
      LHD(N)=NNHD                                                        SP314507
      E=E-CSS(N)                                                         SP314508
      QS=CSS(N)*HSS(NNHD)                                                SP314509
      IF(PHI(N).LT.HSL(NNHD))QS=QS-CSS(N)*(HSL(NNHD)-PHI(N))             SP314510
      IF(PHI(N).GT.HB(NNHD)) QS=QS-CSS(N)*(HB(NNHD)-PHI(N))              SP314511
  138 BL=B/(1.+W*(EL(NIB)+GL(NIB)))                                      SP31460*
      CL=D/(1.+W*(FL(NJB)+GL(NJB)))                                      SP31470
```

```
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-Z*PSP31620
     1HI(NKB)-WELL(N)-RHO*OLD(N)-QR-QS                                  SP31630*
      V(N)=(RES*HMAX-AL*V(NKB)-BL*V(NIB)-CL*V(NJB))/DL                  SP31640*
      GO TO 150                                                         SP31650
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-WELSP31720
     1L(N)-RHO*OLD(N)-QR-QS                                             SP31730*
      V(N)=(RES*HMAX-BL*V(NIB)-CL*V(NJB))/DL                            SP31740*
      GO TO 150                                                         SP31741*
  148 IF(IHDEP.NE.ICHK(13)) GO TO 150                                   SP31742*
      IF (CSS(N).EQ.0.) GO TO 150                                       SP31743*
      NNHD=NNHD+1                                                       SP31744*
  150 CONTINUE                                                          SP31750
      IF(K.EQ.1) GO TO 174                                              SP32231
      IF (T(NKB).EQ.0.) GO TO 174                                       SP322311
      Z=TK(NKB)                                                         SP32232
  174 IF(K.EQ.KO) GO TO 175                                             SP32241
      IF (T(NKA).EQ.0.) GO TO 175                                       SP322411
      SU=TK(N)                                                          SP32242
      QR=0.                                                             SP32260
      QS=0.                                                             SP32261*
      IF (K.NE.KO) GO TO 180                                            SP32270
  180 E=-B-D-F-H-SU-Z-RHO                                               SP32320
C     ---COMPUTE IMPLICIT AND EXPLICIT PARTS OF HEAD-DEPENDENT TERMS--- SP323201
      IF (IHDEP.NE.ICHK(13)) GO TO 188                                  SP323202
      IF (CSS(N).EQ.0.) GO TO 188                                       SP323203
      NNHD=LHD(N)                                                       SP323204
      IF (NNHD.GT.NHD) GO TO 188                                        SP323205
      E=E-CSS(N)                                                        SP323206
      QS=CSS(N)*HSS(NNHD)                                               SP323207
      IF(PHI(N).LT.HSL(NNHD))QS=QS-CSS(N)*(HSL(NNHD)-PHI(N))            SP323208
      IF(PHI(N).GT.HB(NNHD)) QS=QS-CSS(N)*(HB(NNHD)-PHI(N))             SP323209
  188 BL=H/(1.+W*(EL(NIA)+GL(NIA)))                                     SP32330*
      CL=D/(1.+W*(FL(NJB)+GL(NJB)))                                     SP32340
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SU*PHI(NSP32490
     1KA)-ZPHI-WELL(N)-RHO*OLD(N)-QR-QS                                 SP32500*
      V(N)=(RES*HMAX-AL*V(NKA)-BL*V(NIA)-CL*V(NJB))/DL                  SP32510*
      GO TO 200                                                         SP32520
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-ZPHI-WELSP32590
     1L(N)-RHO*OLD(N)-QR-QS                                             SP32600*
      V(N)=(RES*HMAX-BL*V(NIA)-CL*V(NJB))/DL                            SP32610*
  200 CONTINUE                                                          SP32620
     143('_'))                                                          SP32930
  250 FORMAT (2F10.0)                                                   SP32931*
  260 FORMAT (45X,'MAXIMUM ITERATION PARAMETER, WMAX =',F10.7,//,45X,'ACSP32932*
     1CELERATION (OR DAMPENING) FACTOR, HMAX =',F5.2)                   SP32933*
  270 FORMAT (1H0,44X,'INDICATOR FOR WMAX =',F5.2,//,                   SP32934*
     145X,'MAXIMUM ITERATION PARAMETER COMPUTED ACCORDING TO',/,        SP32935*
     245X,'THE ORIGINAL FORMULATION OF TRESCOTT (1975)',//,             SP32936*
     345X,'ACCELERATION (OR DAMPENING) FACTOR HMAX =',F5.2)             SP32937*
  280 FORMAT (1H0,44X,'INDICATOR FOR WMAX =',F5.2,//,                   SP32938*
     140X,'MAXIMUM ITERATION PARAMETER COMPUTED AS THE ABSOLUTE MAX.',/,SP32939*
     240X,'VALUE OVER THE GRID USING FULL FORMULATION OF RHO TERMS',//, SP32940*
     340X,'ACCELERATION (OR DAMPENING) FACTOR HMAX =',F5.2)             SP32941*
  290 FORMAT (1H0,44X,'INDICATOR FOR WMAX =',F5.2,//,                   SP32942*
     140X,'MAXIMUM ITERATION PARAMETER COMPUTED AS THE ARITHMETIC',/,   SP32943*
     240X,'AVERAGE OF THE LOCAL MAXIMA OF WMAX FOR EACH ACTIVE NODE',/, SP32944*
     340X,'USING THE FULL FORMULATION OF RHO TERMS',//,                 SP32945*
     440X,'ACCELERATION (OR DAMPENING) FACTOR HMAX =',F5.2)             SP32946*
      END                                                               SP32947*
```

***** COEF *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NCOF 150
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCCOF 160
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHCOF 170*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                         COF 171*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR               COF 180
      T(I,J,K0)=PERM(I,J)*(PHI(I,J,K0)-BOTTOM(I,J))                     COF 300
      IF (T(I,J,K0).GT.0.) GO TO 5                                      COF 310*
      IF (WELL(I,J,K0).LT.0.) WRITE (6,60) I,J,K0                       COF 320
      PHI(I,J,K0)=1.D30                                                 COF 410
      GO TO 10                                                          COF 411*
    5 IF (IEQN.EQ.ICHK(11)) T(I,J,K0)=PERM(I,J)                         COF 412*
   10 CONTINUE                                                          COF 420
      IF (N3.EQ.1) RETURN                                               COF 430
      IF (IEQN.EQ.ICHK(11).AND.N3.EQ.0) RETURN                          COF 431*
      N1=K0                                                             COF 440
C     *********************                                             COF 490
      ENTRY TCOF(N3)                                                    COF 500*
C     *********************                                             COF 510
```

***** CHECKI *****

```
      SUBROUTINE CHECKI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACHK  10
     1CT,JFLO,FLOW,QRE,CSS,HSS,HB,HSL,HDRT)                            CHK  20*
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0CHK 100
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,CHK 110
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(K0,3), JFLO(NCD,3), FLOCHK 120*
     3W(NCD),QRE(IQ,JQ),CSS(IE,JE,KE),HSS(NHD),HB(NHD),HSL(NHD),       CHK 130*
     4HDRT(NHD)                                                        CHK 131*
C                                                                      CHK 140
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NCHK 150
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCCHK 160
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHCHK 170*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                         CHK 171*
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR               CHK 180
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)                      CHK 190
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT, CHK 200*
     1HDVNT,HDVPT                                                      CHK 201*
      FLXN=0.0                                                          CHK 360
      HDFN=0.                                                           CHK 361*
      HDFP=0.                                                           CHK 362*
      NNHD=0                                                            CHK 363*
      HDRT(1)=0.                                                        CHK 364*
      II=0                                                              CHK 370
      IF (IHDEP.NE.ICHK(13)) GO TO 5                                    CHK 371*
      DO 4 N=2,NHD                                                      CHK 372*
    4 HDRT(N)=0.                                                        CHK 373*
C     ..................................................................CHK 380
      IF (T(I,J,K).EQ.0.) GO TO 218                                     CHK 440*
  210 STOR=STOR+S(I,J,K)*(OLD(I,J,K)-PHI(I,J,K))*AREA                   CHK1090
C                                                                       CHK10901
C     ---COMPUTE FLOW RATES FROM HEAD-DEPENDENT TERMS---                CHK10902
      IF (IHDEP.NE.ICHK(13)) GO TO 220                                  CHK10903
      IF (CSS(I,J,K).EQ.0.)  GO TO 220                                  CHK10904
      NNHD=NNHD+1                                                       CHK10905
      IF (NNHD.GT.NHD) GO TO 220                                        CHK10906
      HDD=PHI(I,J,K)                                                    CHK10907
      IF (HDD.LT.HSL(NNHD)) HDD=HSL(NNHD)                               CHK10908
      IF (HDD.GT.HB(NNHD)) HDD=HB(NNHD)                                 CHK10909
      HDRT(NNHD)=CSS(I,J,K)*(HSS(NNHD)-HDD)*AREA                        CHK10910
      IF (HDRT(NNHD).GT.0.) HDFP=HDFP+HDRT(NNHD)                        CHK10911
```

```
      IF (HDRT(NNHD).LT.0.) HDFN=HDFN+HDRT(NNHD)                        CHK10912
      IF (HSS(NNHD).EQ.HSL(NNHD)) ETFLUX=ETFLUX+HDRT(NNHD)              CHK10913
      GO TO 220                                                         CHK1091
  214 STOR=STOR+S(I,J,K)*(OLD(I,J,K)-PHI(I,J,K))*VOLUME                 CHK1102
C                                                                       CHK11021
C     ---COMPUTE FLOW RATES FROM HEAD-DEPENDENT TERMS---                CHK11022
      IF (IHDEP.NE.ICHK(13)) GO TO 220                                  CHK11023
      IF (CSS(I,J,K).EQ.0.)  GO TO 220                                  CHK11024
      NNHD=NNHD+1                                                       CHK11025
      IF (NNHD.GT.NHD) GO TO 220                                        CHK11026
      HDD=PHI(I,J,K)                                                    CHK11027
      IF (HDD.LT.HSL(NNHD)) HDD=HSL(NNHD)                               CHK11028
      IF (HDD.GT.HB(NNHD)) HDD=HB(NNHD)                                 CHK11029
      HDRT(NNHD)=CSS(I,J,K)*(HSS(NNHD)-HDD)*VOLUME                      CHK11030
      IF (HDRT(NNHD).GT.0.) HDFP=HDFP+HDRT(NNHD)                        CHK11031
      IF (HDRT(NNHD).LT.0.) HDFN=HDFN+HDRT(NNHD)                        CHK11032
      IF (HSS(NNHD).EQ.HSL(NNHD)) ETFLUX=ETFLUX+HDRT(NNHD)              CHK11033
      GO TO 220                                                         CHK11034
  218 IF (IHDEP.NE.ICHK(13)) GO TO 220                                  CHK11035
      IF (CSS(I,J,K).EQ.0.) GO TO 220                                   CHK11036
      NNHD=NNHD+1                                                       CHK11037
  220 CONTINUE                                                          CHK11038
C     ..................................................................CHK1110
      CFLUXT=CFLUXT+CFLUX*DELT                                          CHK1210
      ETFLXT=ETFLXT-ETFLUX*DELT                                         CHK1211*
      HDVNT=HDVNT-HDFN*DELT                                             CHK1212*
      HDVPT=HDVPT+HDFP*DELT                                             CHK1213*
      TOTL1=STORT+QRET+CFLUXT+CHST+FLXPT+HDVPT                          CHK1220*
      TOTL2=CHDT+PUMPT+HDVNT+FLXNT                                      CHK1230*
      SUMR=QREFLX+CFLUX+CHD2+CHD1+PUMP+FLUXS+STOR+HDFP+HDFN             CHK1240*
      DIFF=TOTL2-TOTL1                                                  CHK1250
     2L2,DIFF,PERCNT                                                    CHK1390
      IF (NCH.EQ.0) GO TO 231                                           CHK1400*
      WRITE (6,270)                                                     CHK1410
      WRITE (6,280) ((JFLO(I,J),J=1,3),FLOW(I),I=1,NCH)                 CHK1420
C                                                                       CHK1430
C     ---PRINT OUT FLOW RATES FROM HEAD-DEPENDENT SOURCES AND SINKS---  CHK14301
  231 IF(IHDEP.NE.ICHK(13)) GO TO 240                                   CHK14302
      IRR=0                                                             CHK14303
      NNHD=0                                                            CHK14304
      II=0                                                              CHK14305
      DO 234 K=1,KO                                                     CHK14306
      DO 234 I=2,I1                                                     CHK14307
      DO 234 J=2,J1                                                     CHK14308
      IF(CSS(I,J,K).EQ.0.) GO TO 234                                    CHK14309
      NNHD=NNHD+1                                                       CHK14310
      IF(NNHD.GT.NHD) GO TO 240                                         CHK14311
      IF(HDRT(NNHD).LE.0.) GO TO 232                                    CHK14312
      II=II+1                                                           CHK14313
      JFLO(II,1)=K                                                      CHK14314
      JFLO(II,2)=I                                                      CHK14315
      JFLO(II,3)=J                                                      CHK14316
      FLOW(II)=HDRT(NNHD)                                               CHK14317
      GO TO 234                                                         CHK14318
  232 IR=NHD-IRR                                                        CHK14319
      JFLO(IR,1)=K                                                      CHK14320
      JFLO(IR,2)=I                                                      CHK14321
      JFLO(IR,3)=J                                                      CHK14322
      FLOW(IR)=HDRT(NNHD)                                               CHK14323
      IRR=IRR+1                                                         CHK14323
  234 CONTINUE                                                          CHK14324
```

```
      IF (II.EQ.0) GO TO 236                                            CHK14325
      WRITE(6,300)                                                      CHK14326
      WRITE(6,280) ((JFLO(I,J),J=1,3),FLOW(I),I=1,II)                   CHK14327
      WRITE(6,310) HDFP,HDVPT                                           CHK14328
  236 IF(IRR.EQ.0.) GO TO 240                                           CHK14329
      WRITE(6,320)                                                      CHK14330
      WRITE(6,280) ((JFLO(I,J),J=1,3),FLOW(I),I=IR,NHD)                 CHK14331
      WRITE(6,330) HDFN,HDVNT                                           CHK14332
C     ---COMPUTE VERTICAL FLOW---                                       CHK1440
  240 X=0.                                                              CHK1450
      Y=0.                                                              CHK1460
      IF (K0.EQ.1) RETURN                                               CHK1470
      DO 250 I=2,I1                                                     CHK1480
      DO 250 J=2,J1                                                     CHK1490
      IF(T(I,J,1).EQ.0..OR.T(I,J,2).EQ.0.) GO TO 245                    CHK1491*
      IF(S(I,J,1).LT.0..AND.S(I,J,2).LT.0.) GO TO 245                   CHK1492*
      X=X+(PHI(I,J,1)-PHI(I,J,2))*TK(I,J,1)*DELX(J)*DELY(I)             CHK1500
  245 IF(T(I,J,K1).EQ.0..OR.T(I,J,K0).EQ.0.) GO TO 250                  CHK1501*
      IF(S(I,J,K1).LT.0..AND.S(I,J,K0).LT.0.) GO TO 250                 CHK1502*
      Y=Y+(PHI(I,J,K1)-PHI(I,J,K0))*TK(I,J,K1)*DELX(J)*DELY(I)          CHK1520*
  250 CONTINUE                                                          CHK1521*
      WRITE (6,290) Y,X                                                 CHK1540
     3,'STORAGE =',F20.2,35X,'CONSTANT FLUX =',F20.4/26X,'RECHARGE =',F2CHK1660
     40.2,41X,'PUMPING =',F20.4/21X,'CONSTANT FLUX =',F20.2,30X,'ET. ANDCHK1670*
     5/OR SPRINGS =',F20.4/21X,'CONSTANT HEAD =',F20.2,34X,'CONSTANT HEACHK1680*
     6D:'/27X,'LEAKAGE =',F20.2,46X,'IN =',F20.4/21X,'TOTAL SOURCES =',FCHK1690
     720.2,45X,'OUT =',F20.4/96X,'LEAKAGE:'/20X,'DISCHARGES:',45X,'FROM CHK1700
     8PREVIOUS PUMPING PERIOD =',F20.4/20X,11('-'),68X,'TOTAL =',F20.4/1CHK1710
     96X,'ET. AND/OR SPRINGS =',F20.2/21X,'CONSTANT HEAD =',F20.2,36X,'SCHK1720*
     SUM OF RATES =',F20.4/19X'QUANTITY PUMPED =',F20.2/27X,'LEAKAGE =',CHK1730
     115.7,'       POSITIVE UPWARD')                                    CHK1810
  300 FORMAT('0FLOW RATES FROM HEAD-DEPENDENT SOURCES:',/,' ',39('-')//'CHK1811*
     1 ',3(9X,'K',4X,'I',4X,'J',5X,'RATE (L**3/T)')/' ',3(9X,'-',4X,'-',CHK1812*
     24X,'-',5X,13('-'))/)                                              CHK1813*
  310 FORMAT(1H0,15X,'TOTAL FLOW RATE FROM HEAD-DEPENDENT SOURCES =',G18CHK1814*
     1.7,10X,'TOTAL VOLUME =',G18.7)                                    CHK1815*
  320 FORMAT('0FLOW RATES FROM HEAD-DEPENDENT SINKS:',/,' ',37('-')//' 'CHK1816*
     1,3(9X,'K',4X,'I',4X,'J',5X,'RATE (L**3/T)')/' ',3(9X,'-',4X,'-',4XCHK1817*
     2,'-',5X,13('-'))/)                                                CHK1818*
  330 FORMAT(1H0,17X,'TOTAL FLOW RATE FROM HEAD-DEPENDENT SINKS =',G18.7CHK1819*
     1,10X,'TOTAL VOLUME =',G18.7)                                      CHK1820*
      END                                                               CHK1820-
```

***** PRINTAI *****

```
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NPRN 130
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NCPRN 140
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK,IEQN,IHPRN 150*
     3DEP,ITL,NHD,IE,JE,KE,NCD                                          PRN 151*
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANKPRN 160
```

***** BLOCK DATA *****

```
      DATA ICHK/'DRAW','HEAD','MASS','DK1','DK2','WATE','RECH','PUN1','PBLK 130
     1UN2','ITKR','EQN3','ITLR','HDEP'/                                 BLK 140*
```

APPENDIX V

Example of Input to Modified Model

```
//STEP1 EXEC FORTRUN,PROC=HEDEP,REGION=490K,ULIB='VC4E91L.THREDEE'
//GO.SYSIN DD *
**** CHECKOUT SIMULATION OF 3-D FINITE-DIFFERENCE MODEL (TRESCOTT), MODIFIED TO
INCLUDE HEAD-DEPENDENT SOURCE/SINK TERMS ****
         8         8         3        60        36                  12
    HEAD MASS                                             HDEP
         1         2       .01         5
0
0
0
0
0
     1.E4          1
0
                 115       115       115       115       115       115
0
0
0
0
                  90        90        90        90        90        90
0
     1.E4          1
0
                 115       115       115       115       115       115
0
0
0
0
                  90        90        90        90        90        90
0
     1.E4          1
0
                 115       115       115       115       115       115
0
0
0
0
                  90        90        90        90        90        90
0
       1.          1
0
   0  -1  -1  -1  -1  -1  -1   0
0
0
0
0
   0  -1  -1  -1  -1  -1  -1   0
0
       1.          1
0
   0  -1  -1  -1  -1  -1  -1   0
0
0
0
```

```
C
   0  -1   -1   -1   -1   -1   -1    0
C
        1.          1
C
   0  -1   -1   -1   -1   -1   -1    0
C
C
C
C
   0  -1   -1   -1   -1   -1   -1    0
C
     8.659                                   1.          1.       .0001
    25.976                                   1.          1.       .0001
    17.317                                   1.          1.       .0001
 3.624E-08         1       **** CSS ARRAY: LAYER 1 ****
C
C
           1
           1
           1
           1
C
C
 3.624E-08         1       **** CSS ARRAY: LAYER 2 ****
C
C
               1
               1
               1
               1
C
C
 3.624E-08         1       **** CSS ARRAY: LAYER 3 ****
C
C
                   1
                   1
                   1
                   1
C
C
      110.       100.       100.       HE(N)   HSS(N)   HSL(N)
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
      110.       100.       100.
       1E4
       1E4
      100.
        3.       1.2                   WMAX  HMAX
         1         0          0          1          1       1.0        24.
/*
//
```

APPENDIX VI

Example of Output from Modified Model

```
**** CHECKOUT SIMULATION OF 3-D FINITE-DIFFERENCE MODEL (TRESCOTT), MODIFIED TO INCLUDE HEAD-DEPENDENT SOURCE/SINK TERMS ****

                                                  NUMBER OF ROWS =    8
                                               NUMBER OF COLUMNS =    8
                                                NUMBER OF LAYERS =    3

                           MAXIMUM PERMITTED NUMBER OF ITERATIONS =   60

                                   NUMBER OF CONSTANT HEAD NODES =   36

                        NUMBER OF HEAD-DEPENDENT SOURCE-SINK NODES =   12

SIMULATION OPTIONS:        HEAD    MASS                                                              HDEP

                                          WORDS OF VECTOR Y USED =   3493

                                       NUMBER OF PUMPING PERIODS =    1
                                   TIME STEPS BETWEEN PRINTOUTS =    2

                                       ERROR CRITERIA FOR CLOSURE =   .10000001-01
```

STARTING HEAD MATRIX, LAYER 1

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	115.0	115.0	115.0	115.0	115.0	115.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	90.0	90.0	90.0	90.0	90.0	90.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

STARTING HEAD MATRIX, LAYER 2

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	115.0	115.0	115.0	115.0	115.0	115.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	90.0	90.0	90.0	90.0	90.0	90.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

STARTING HEAD MATRIX, LAYER 3

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	115.0	115.0	115.0	115.0	115.0	115.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	90.0	90.0	90.0	90.0	90.0	90.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

STORAGE COEFFICIENT MATRIX, LAYER 1

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

STORAGE COEFFICIENT MATRIX, LAYER 2

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

STORAGE COEFFICIENT MATRIX, LAYER 3

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

TRANSMISSIVITY = 8.659000 FOR LAYER 1

DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORS FOR LAYER 1
X = 1.000000
Y = 1.000000
Z = .1000000E-03

TRANSMISSIVITY = 25.97600 FOR LAYER 2

DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORS FOR LAYER 2
X = 1.000000
Y = 1.000000
Z = .1000000E-03

TRANSMISSIVITY = 17.31700 FOR LAYER 3

DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORS FOR LAYER 3
X = 1.000000
Y = 1.000000
Z = .1000000E-03

HEAD-DEPENDENT LEAKANCES MATRIX, LAYER 1

1	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
3	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

HEAD-DEPENDENT LEAKANCES MATRIX, LAYER 2

1	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
3	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

HEAD-DEPENDENT LEAKANCES MATRIX, LAYER 3

1	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
3	0.0	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.36240E-07	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

THRESHOLD HEADS FOR HEAD-DEPENDENT SOURCE-SINK TERMS

LAYER K	ROW I	COLUMN J	UPPER (HD)	MIDDLE (HSS)	LOWER (HSL)
1	3	3	110.00	100.00	100.00
1	4	3	110.00	100.00	100.00
1	5	3	110.00	100.00	100.00
1	6	3	110.00	100.00	100.00
2	3	4	110.00	100.00	100.00
2	4	4	110.00	100.00	100.00
2	5	4	110.00	100.00	100.00
2	6	4	110.00	100.00	100.00
3	3	5	110.00	100.00	100.00
3	4	5	110.00	100.00	100.00
3	5	5	110.00	100.00	100.00
3	6	5	110.00	100.00	100.00

DELX = 10000.00

DELY = 10000.00

DELZ = 100.0000

SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE

INDICATOR FOR WMAX = 3.00

MAXIMUM ITERATION PARAMETER COMPUTED AS THE ARITHMETIC AVERAGE OF THE LOCAL MAXIMA OF WMAX FOR EACH ACTIVE NODE USING THE FULL FORMULATION OF RHO TERMS

ACCELERATION (OR DAMPENING) FACTOR WMAX = 1.20

5 ITERATION PARAMETERS: 0.0 0.8126363D+00 0.9648948D+00 0.9934226D+00 0.9987676D+00

```
PUMPING PERIOD NO.   1:      1.00 DAYS
--------------------------------------

 NUMBER OF TIME STEPS=     1

       DELT IN HOURS =    24.000

 MULTIPLIER FOR DELT =     1.000

                0 WELLS
              ---------

K          I          J    PUMPING RATE
```

```
| TIME STEP NUMBER = 1 |

SIZE OF TIME STEP IN SECONDS=      86400.00

TOTAL SIMULATION TIME IN SECONDS=    86400.00
                        MINUTES=     1440.00
                          HOURS=       24.00
                           DAYS=        1.00
                          YEARS=        0.00

DURATION OF CURRENT PUMPING PERIOD IN DAYS=    1.00
                                    YEARS=    0.00
```

CUMULATIVE MASS BALANCE:	L**3
SOURCES:	
STORAGE =	0.0
RECHARGE =	0.0
CONSTANT FLUX =	0.0
CONSTANT HEAD =	144628864.
LEAKAGE =	0.0
TOTAL SOURCES =	144628864.
DISCHARGES:	
ET. AND/OR SPRINGS =	13106351.0
CONSTANT HEAD =	131013952.
QUANTITY PUMPED =	0.0
LEAKAGE =	0.0
TOTAL DISCHARGE =	144120288.
DISCHARGE-SOURCES =	-508576.00
PER CENT DIFFERENCE =	-0.35

RATES FOR THIS TIME STEP:	L**3/T
STORAGE =	0.0
RECHARGE =	0.0
CONSTANT FLUX =	0.0
PUMPING =	0.0
ET. AND/OR SPRINGS =	-151.6935
CONSTANT HEAD:	
IN =	1673.9453
OUT =	-1516.3652
LEAKAGE:	
FROM PREVIOUS PUMPING PERIOD =	0.0
TOTAL =	0.0
SUM OF RATES =	5.8862

FLOW RATES TO CONSTANT HEAD NODES:

K	I	J	RATE (L**3/T)	K	I	J	RATE (L**3/T)	K	I	J	RATE (L**3/T)
1	2	2	45.82071	1	2	3	47.85675	1	2	4	48.06738
1	2	5	47.44109	1	2	6	45.31520	1	2	7	44.59309
1	7	2	-42.04755	1	7	3	-41.94511	1	7	4	-41.93355
1	7	5	-42.06548	1	7	6	-42.29732	1	7	7	-42.44827
2	2	2	137.4533	2	2	3	143.0059	2	2	4	144.3086
2	2	5	142.4294	2	2	6	135.9427	2	2	7	133.7762
2	7	2	-126.1377	2	7	3	-125.8305	2	7	4	-125.7961
2	7	5	-126.1918	2	7	6	-126.8876	2	7	7	-127.3403
3	2	2	91.63237	3	2	3	95.24348	3	2	4	96.11084
3	2	5	95.13832	3	2	6	90.62814	3	2	7	89.18320
3	7	2	-84.08986	3	7	3	-83.88535	3	7	4	-83.86249

3	7	5	-84.12613	3	7	6	-84.59007	3	7	7	-84.89189

FLOW RATES FROM HEAD-DEPENDENT SINKS:

K	I	J	RATE (L**3/T)	K	I	J	RATE (L**3/T)	K	I	J	RATE (L**3/T)
3	6	5	.0	3	5	5	.0	3	4	5	-16.34564
3	3	5	-34.44994	2	6	4	.0	2	5	4	.0
2	4	4	-16.12108	2	3	4	-34.22699	1	6	3	.0
1	5	3	.0	1	4	3	-16.21951	1	3	3	-34.33072

TOTAL FLOW RATE FROM HEAD-DEPENDENT SINKS = -151.6939 TOTAL VOLUME = .1310635E+08

FLOW TO TOP LAYER = .9779500 FLOW TO BOTTOM LAYER = -25.08482 POSITIVE UPWARD

MAXIMUM HEAD CHANGE FOR EACH ITERATION:

52.3669	30.2570	35.4181	32.7166	17.3975	1.4848	0.9067	0.4888	0.3863	0.1270
0.0172	0.0098								

TIME STEP : 1

ITERATIONS: 11

HEAD MATRIX, LAYER 1

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	115.00	115.00	115.00	115.00	115.00	115.00	0.0
3	0.0	109.71	109.47	109.45	109.52	109.77	109.85	0.0
4	0.0	104.64	104.48	104.45	104.52	104.71	104.79	0.0
5	0.0	99.72	99.68	99.67	99.70	99.78	99.82	0.0
6	0.0	94.86	94.84	94.84	94.86	94.88	94.90	0.0
7	0.0	90.00	90.00	90.00	90.00	90.00	90.00	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

HEAD MATRIX, LAYER 2

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	115.00	115.00	115.00	115.00	115.00	115.00	0.0
3	0.0	109.71	109.49	109.44	109.52	109.77	109.85	0.0
4	0.0	104.64	104.49	104.45	104.52	104.71	104.79	0.0
5	0.0	99.72	99.68	99.67	99.70	99.78	99.82	0.0
6	0.0	94.86	94.84	94.84	94.86	94.88	94.90	0.0
7	0.0	90.00	90.00	90.00	90.00	90.00	90.00	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

HEAD MATRIX, LAYER 3

1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	115.00	115.00	115.00	115.00	115.00	115.00	0.0
3	0.0	109.71	109.50	109.45	109.51	109.77	109.85	0.0
4	0.0	104.64	104.49	104.45	104.51	104.71	104.79	0.0
5	0.0	99.72	99.68	99.67	99.70	99.78	99.82	0.0
6	0.0	94.86	94.84	94.84	94.86	94.88	94.90	0.0
7	0.0	90.00	90.00	90.00	90.00	90.00	90.00	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0