DOCUMENTATION OF FINITE-DIFFERENCE MODEL

FOR

SIMULATION OF THREE-DIMENSIONAL GROUND-WATER FLOW

By

Peter C. Trescott

U.S. Geological Survey Open File Report 75-438

September 1975

Reprint July 1976

(Supplemented by OFR 76-591)

ACKNOWLEDGMENT

The efficiency of the program has been improved significantly by Steven P. Larson who added the option of storing input arrays on disk and modified the solution routine to use only single-subscript arrays.

Requests for copies of this documentation and the card deck should be directed to:

Ralph N. Eicher, Chief
Office of Teleprocessing, M.S. 805
U.S. Geological Survey
Reston, Virginia 22092

CONTENTS

FIGURES

TABLES

INTRODUCTION

This report documents a finite-difference model for simulation of ground-water flow in three dimensions. Although the techniques for constructing numerical models of ground-water flow in three dimensions have been available in the petroleum literature for some time, the large computer-memory requirements for simulation of realistic ground-water problems have discouraged development of such models. To minimize the core requirements for these simulations, Bredehoeft and Pinder (1970) described a quasi three-dimensional model which is a sequence of two-dimensional ground-water flow models coupled by terms representing flow through intervening confining beds. In practice there has been some difficulty getting convergence with this approach. The fully three-dimensional model described in this report can be reduced to a quasi three-dimensional model in terms of the equations being solved and computer-memory requirements, yet converges to a solution much faster because all equations are solved simultaneously.

The iterative numerical technique used to solve the set of simultaneous finite-difference equations is the strongly implicit procedure (SIP) originally described by Stone (1968) for problems in two dimensions and extended to three dimensions by Weinstein, Stone and Kwan (1969). Weinstein and others (1969) claim that SIP converges faster than the iterative alternating-direction implicit procedure (ADI) (used for solution in two dimensions by Bredehoeft and Pinder, 1970) and that SIP is less subject to roundoff errors than ADI.

THEORETICAL DEVELOPMENT

Ground-water flow equation

The flow of ground water in a porous medium in three dimensions may be expressed as (Jacob, 1950, Cooper, 1966)

$$\nabla^2 h = \frac{S_s}{K}\frac{\partial h}{\partial t} \qquad (1)$$

in which h is hydraulic head (L);

S_s is specific storage (L^{-1});

K is hydraulic conductivity ($L\ T^{-1}$).

Permitting hydraulic conductivity to be heterogeneous and anisotropic and adding a source term, equation 1 becomes

$$\nabla\cdot(K_{ij}\frac{\partial h}{\partial x_j}) = S_s\frac{\partial h}{\partial t} + W(x,y,z,t) \qquad (2)$$

in which $W(x,y,z,t)$ is a volumetric flux per unit volume (T^{-1}). Assuming that the coordinate axes x,y and z are alined with the principal directions of the hydraulic conductivity tensor, the cross-product terms drop out of equation 2 and it becomes in expanded form

$$\frac{\partial}{\partial x}(K_{xx}\frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}(K_{yy}\frac{\partial h}{\partial y}) + \frac{\partial}{\partial z}(K_{zz}\frac{\partial h}{\partial z}) = S_s\frac{\partial h}{\partial t} + W(x,y,z,t) \qquad (3)$$

in which

K_{xx}, K_{yy}, K_{zz} are the principal components of the hydraulic conductivity tensor ($L\ T^{-1}$).

In the finite-difference simulator, it is often convenient to represent a hydraulic unit by one layer of nodes. For this approach, equation 3 is multiplied by b, the thickness of the hydraulic unit giving

$$\frac{\partial}{\partial x}(T_{xx}\frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}(T_{yy}\frac{\partial h}{\partial y}) + \frac{\partial}{\partial z}(bK_{zz}\frac{\partial h}{\partial z}) = S'\frac{\partial h}{\partial t} + bW(x,y,z,t) \qquad (4)$$

in which T_{xx}, T_{yy} are the principal components of the transmissivity tensor (L^2T^{-1});

S' is the storage coefficient (dimensionless).

Although the model is designed to solve equation 4, it will solve equation 3 by substituting hydraulic conductivity, specific storage, and W(x,y,z,t) for transmissivity, storage coefficient, and bW(x,y,z,t), respectively.

Finite-difference approximation

In order to solve equation 3 or 4 for a heterogeneous, anisotropic porous medium with irregular boundaries, one approach is to subdivide the region into blocks in which the medium properties are assumed to be uniform. The continuous derivatives in equation 3 or 4 are replaced by finite-difference approximations for the derivatives at a point (the node at the center of the block). The result is N equations in N unknowns (head values at the nodes) where N is the number of blocks representing the porous medium.

Utilizing a block-centered, finite-difference grid in which variable grid spacing is permitted (figure 1), equation 4 may be approximated as

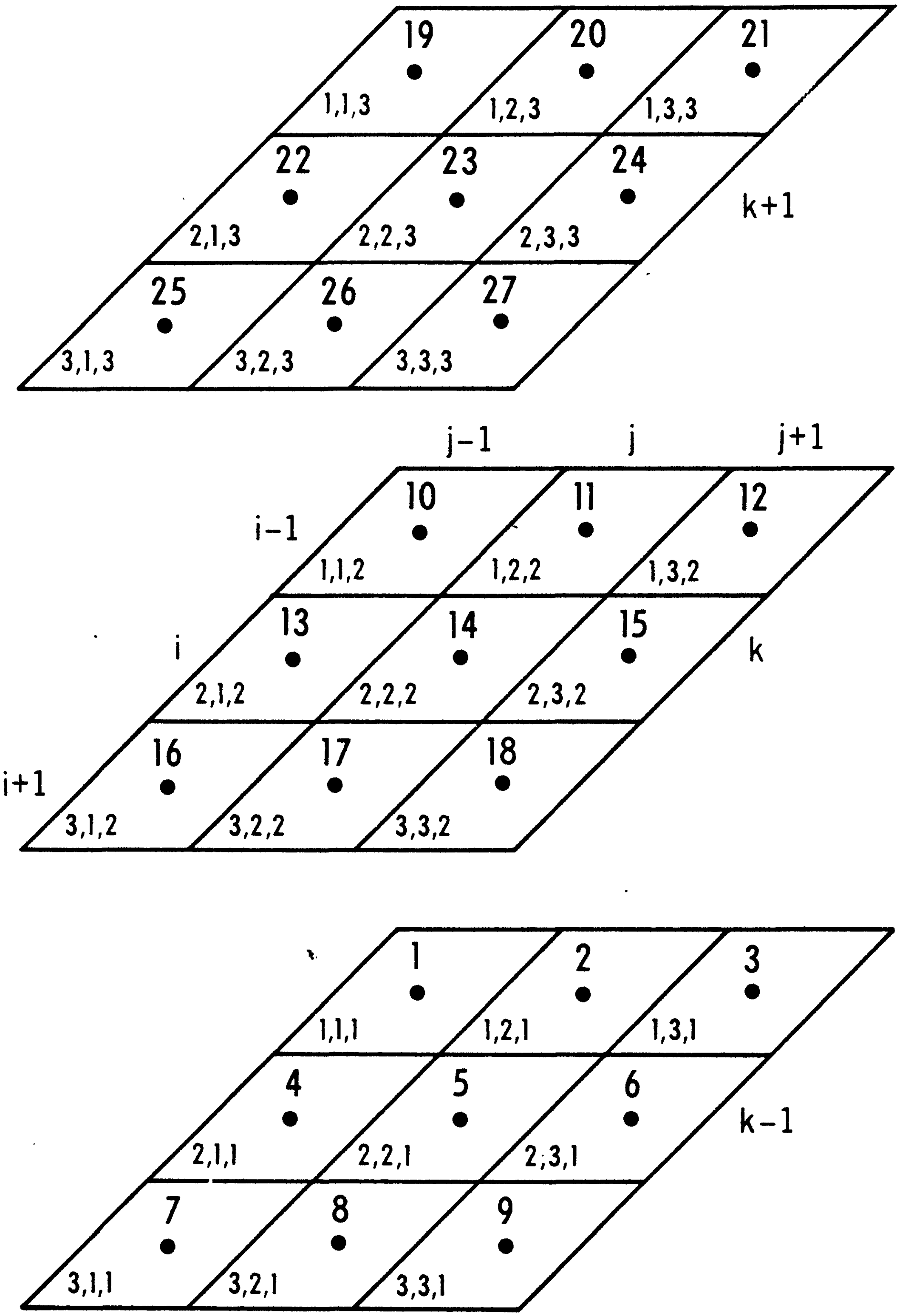

Figure 1.--Index scheme for finite-difference grid and the normal ordering of nodes for 3 X 3 X 3 problem. The numerical values of the indices (i,j,k) for this problem are given in the lower left-hand corner of each block.

$$\frac{1}{\Delta x_j} \left[\left(T_{xx} \frac{\partial h}{\partial x}\right)_{i,j+\frac{1}{2},k} - \left(T_{xx} \frac{\partial h}{\partial x}\right)_{i,j-\frac{1}{2},k} \right] +$$

$$\frac{1}{\Delta y_i} \left[\left(T_{yy} \frac{\partial h}{\partial y}\right)_{i+\frac{1}{2},j,k} - \left(T_{yy} \frac{\partial h}{\partial y}\right)_{i-\frac{1}{2},j,k} \right] +$$

$$\frac{1}{\Delta z_k} \left[\left(bK_{zz} \frac{\partial h}{\partial z}\right)_{i,j,k+\frac{1}{2}} - \left(bK_{zz} \frac{\partial h}{\partial z}\right)_{i,j,k-\frac{1}{2}} \right] =$$

$$\frac{S'_{i,j,k}}{\Delta t} \left(h_{i,j,k} - \hat{h}_{i,j,k}\right) + bW_{i,j,k} \qquad (5)$$

in which Δx_j is the space increment in the x direction for column j (L);

Δy_i is the space increment in the y direction for row i (L);

Δz_k is the space increment in the z direction for layer k (L);

Δt is the time increment (T);

i is the index in the y dimension;

j is the index in the x dimension;

k is the index in the z dimension;

$\hat{h}_{i,j,k}$ is the hydraulic head at the previous time step (L).

The final approximation for equation 4 is

$$\frac{1}{\Delta x_j}\left\{\left[T_{xx(i,j+\frac{1}{2},k)}\frac{(h_{i,j+1,k}-h_{i,j,k})}{\Delta x_{j+\frac{1}{2}}}\right]-\left[T_{xx(i,j-\frac{1}{2},k)}\frac{(h_{i,j,k}-h_{i,j-1,k})}{\Delta x_{j-\frac{1}{2}}}\right]\right\}+$$

$$\frac{1}{\Delta y_i}\left\{\left[T_{yy(i+\frac{1}{2},j,k)}\frac{(h_{i+1,j,k}-h_{i,j,k})}{\Delta y_{i+\frac{1}{2}}}\right]-\left[T_{yy(i-\frac{1}{2},j,k)}\frac{(h_{i,j,k}-h_{i-1,j,k})}{\Delta y_{i-\frac{1}{2}}}\right]\right\}+$$

$$\frac{1}{\Delta z_k}\left\{\left[(bK_{zz})_{i,j,k+\frac{1}{2}}\frac{(h_{i,j,k+1}-h_{i,j,k})}{\Delta z_{k+\frac{1}{2}}}\right]-\left[(bK_{zz})_{i,j,k-\frac{1}{2}}\frac{(h_{i,j,k}-h_{i,j,k-1})}{\Delta z_{k-\frac{1}{2}}}\right]\right\}$$

$$=\frac{S'_{i,j,k}}{\Delta t}(h_{i,j,k}-\hat{h}_{i,j,k})+bW_{i,j,k}\;. \qquad (6)$$

Following a convention similar to that introduced by Stone (1968), the notation in equation 6 may be simplified by writing

$$F_{i,j,k}(h_{i,j+1,k}-h_{i,j,k})-D_{i,j,k}(h_{i,j,k}-h_{i,j-1,k})\;+$$

$$H_{i,j,k}(h_{i+1,j,k}-h_{i,j,k})-B_{i,j,k}(h_{i,j,k}-h_{i-1,j,k})\;+$$

$$S_{i,j,k}(h_{i,j,k+1}-h_{i,j,k})-Z_{i,j,k}(h_{i,j,k}-h_{i,j,k-1})$$

$$=\frac{S'_{i,j,k}}{\Delta t}(h_{i,j,k}-\hat{h}_{i,j,k})+bW_{i,j,k}\;, \qquad (7)$$

in which

$$B_{i,j,k} = \left[\frac{2T_{yy[i,j,k]}T_{yy[i-1,j,k]}}{T_{yy[i,j,k]}\,\Delta y_{i-1} + T_{yy[i-1,j,k]}\,\Delta y_{i}}\right] / \Delta y_{i} \qquad (8a)$$

and the terms in brackets is the harmonic mean of

$$\frac{T_{yy[i,j,k]}}{\Delta y_{i}}, \quad \frac{T_{yy[i-1,j,k]}}{\Delta y_{i-1}} .$$

Similarly,

$$D_{i,j,k} = \left[\frac{2T_{xx[i,j,k]}\,T_{xx[i,j-1,k]}}{T_{xx[i,j,k]}\Delta x_{j-1} + T_{xx[i,j-1,k]}\,\Delta x_{j}}\right] / \Delta x_{j}; \qquad (8b)$$

$$F_{i,j,k} = \left[\frac{2T_{xx[i,j,k]}\,T_{xx[i,j+1,k]}}{T_{xx[i,j,k]}\,\Delta x_{j+1} + T_{xx[i,j+1,k]}\Delta x_{j}}\right] / \Delta x_{j}; \qquad (8c)$$

$$H_{i,j,k} = \left[\frac{2T_{yy[i+1,j,k]}\,T_{yy[i,j,k]}}{T_{yy[i,j,k]}\,\Delta y_{i+1} + T_{yy[i+1,j,k]}\,\Delta y_{i}}\right] / \Delta y_{i}; \qquad (8d)$$

$$S_{i,j,k} = \left[\frac{2(bK_{zz})_{i,j,k+1}(bK_{zz})_{i,j,k}}{(bK_{zz})_{i,j,k}\Delta z_{k+1} + (bK_{zz})_{i,j,k+1}\Delta z_{k}}\right] / \Delta z_{k}; \qquad (8e)$$

$$Z_{i,j,k} = \left[\frac{2(bK_{zz})_{i,j,k-1}(bK_{zz})_{i,j,k}}{(bK_{zz})_{i,j,k}\Delta z_{k-1} + (bK_{zz})_{i,j,k-1}\Delta z_{k}}\right] / \Delta z_{k}. \qquad (8f)$$

Use of the harmonic mean 1) insures continuity across block boundaries at steady state for an irregular grid and 2) makes the appropriate coefficients zero at no-flow boundaries.

If the upper hydrologic unit is under water-table conditions, the specific yield is inserted for the storage coefficient and the transmissivities in equations 8 are defined as a function of the head for the previous iteration. As an example

$$T^n_{xx\ (i,j,k)} = K_{xx(i,j,k)} b^{n-1}_{i,j,k}$$

in which $b^{n-1}_{i,j,k}$ is the saturated thickness of the upper hydrologic unit at iteration n-1;

n is the iteration index.

The notation in equation 7 can be simplified by eliminating all subscripts not including a'+1' or '-1'. After rearranging equation 7 by placing all of the unknowns on the left-hand side, it becomes

$$Bh_{i-1} + Dh_{j-1} + Eh + Fh_{j+1} + Hh_{i+1} + Sh_{k+1} + Zh_{k-1} = Q \qquad (9)$$

in which

$$E = -\ (B + D + F + H + S + Z + \frac{S'}{\Delta t});$$

$$Q = -\frac{S'}{\Delta t}\ \hat{h} + bW.$$

Source Term

The source term W(x,y,z,t) can include well discharge and recharge from precipitation. In the model the source term is computed as

$$(bW)_{i,j,k} = \frac{Q_{w[i,j,k]}}{\Delta x_j \ \Delta y_i} - q_{re[i,j,k]}$$

in which

$Q_{w[i,j,k]}$	is the well discharge (L^3T^{-1});
$q_{re[i,j,k]}$	is the volumetric flux per unit area of the uppermost hydraulic unit (LT^{-1}). (Adding this term to other layers will require modifications to the code.)

To solve equation 3 instead of 4, divide $Q_{w[i,j,k]}$ and $q_{re[i,j,k]}$ by Δz_k for data input.

Numerical Solution by the Strongly Implicit Procedure

The following development follows that given in Weinstein, Stone and Kwan (1969). The nodes (i,j,k) may be ordered as in figure 1 so that j is swept through first (j = 1,2,...,J), i is swept through second (i = 1,2,...,I) and k last (k = 1,2,...,K'). Ordering the set of equations 9 in the same manner, they may be represented by the matrix equation

$$\bar{\bar{A}}\,\bar{h} = \bar{Q} \tag{10}$$

in which

$$\bar{h} = \begin{bmatrix} h_{1,1,1} \\ h_{1,2,1} \\ \vdots \\ h_{1,J,1} \\ h_{2,1,1} \\ h_{2,2,1} \\ \vdots \\ h_{2,J,1} \\ \vdots \\ \vdots \\ h_{I,J,K'} \end{bmatrix}$$

and I is the number of rows;

J is the number of columns;

K' is the number of layers.

$\bar{Q}$ has the same form as $\bar{h}$, and the matrix $\bar{\bar{A}}$ is shown in figure 2. Equation 10 may be solved directly by factoring $\bar{\bar{A}}$ into lower $\bar{\bar{L}}$ and upper $\bar{\bar{U}}$ triangular matrices. However, $\bar{\bar{L}}$ has non-zero elements in all the diagonals from Z to E and $\bar{\bar{U}}$ has non-zero elements in all of the diagonals from E to S. Consequently, solution by Gaussian elimination for problems of practical size requires excessive computation time and computer storage.

To include as many zero diagonals in $\bar{\bar{L}}$ and $\bar{\bar{U}}$ as possible, a modifying matrix $\bar{\bar{B}}$ can be added to $\bar{\bar{A}}$ such that $(\overline{\overline{A+B}})$ factors into $\bar{\bar{L}}$ and $\bar{\bar{U}}$ as shown in figures 3 and 4 . With this modification, equation 10 becomes

$$(\overline{\overline{A + B}})\ \bar{h} = \bar{Q} + \bar{\bar{B}}\bar{h} \tag{11}$$

Equation 11 can be solved readily if the right-hand side is known. This leads to an iterative scheme defined by the following equation:

$$(\overline{\overline{A + B}})\ \bar{h}^{n} = \bar{Q} + \bar{\bar{B}}\bar{h}^{n-1} \tag{12}$$

in which n is an iteration index. To reduce roundoff errors, equation 12 is put in residual form by adding and subtracting $\bar{\bar{A}}\bar{h}^{n-1}$ to the right hand side:

$$(\overline{\overline{A + B}})\ \bar{\xi}^{n} = \bar{R}^{n-1} \tag{13}$$

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

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

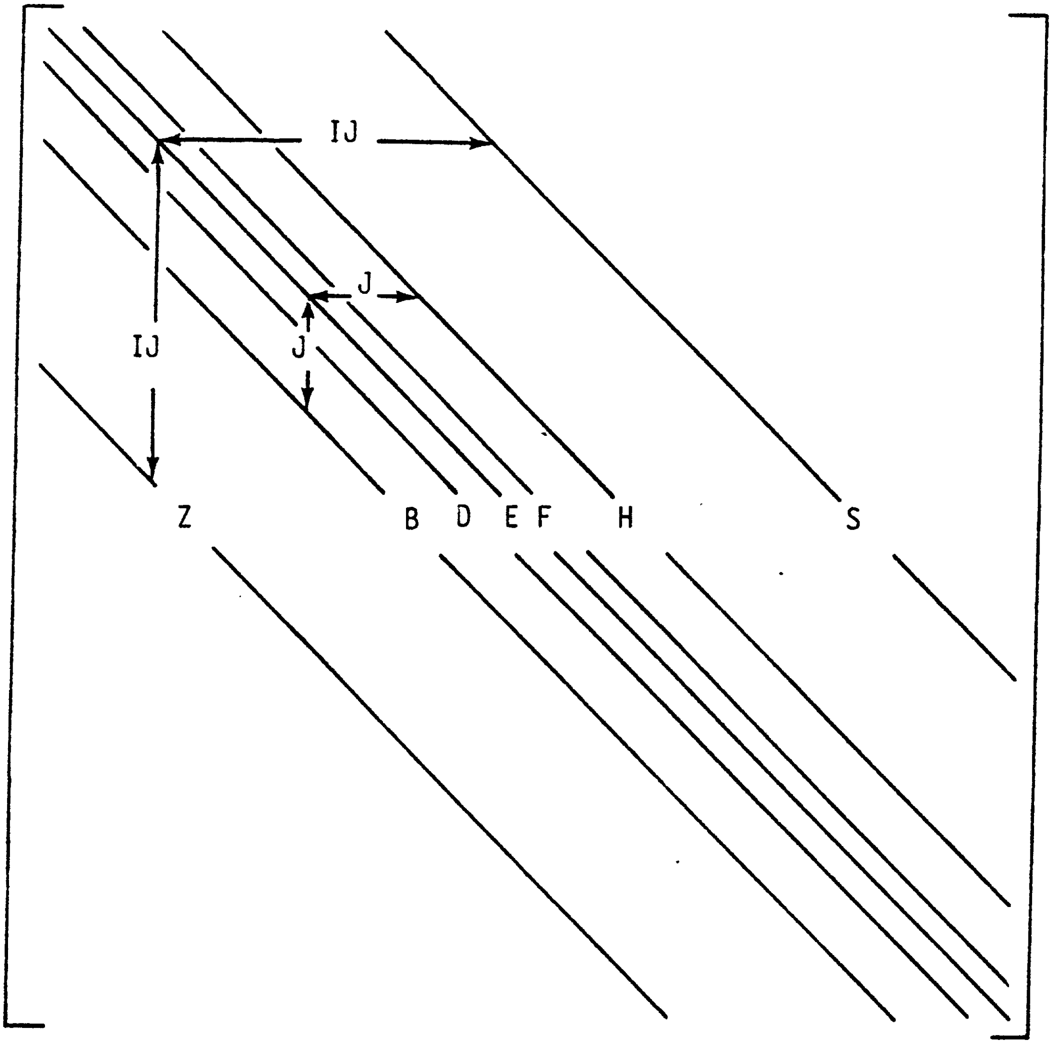

Figure 2.--Matrix $\overset{=}{A}$.

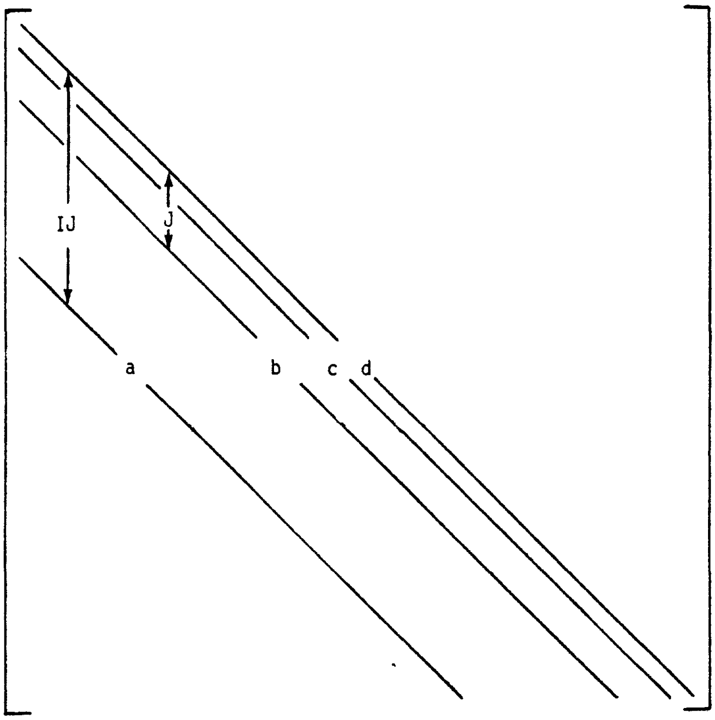

Figure 3.--Lower triangular factor $\bar{\bar{L}}$ of $(\overline{\overline{A+B}})$.

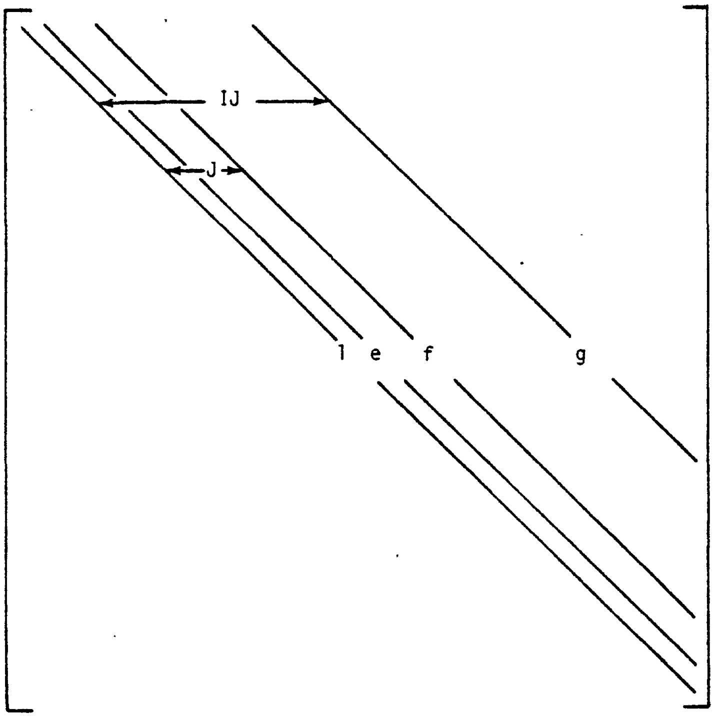

Figure 4.--Upper triangular factor $\bar{\bar{U}}$ of $(\overline{\overline{A+B}})$.

$\bar{R}^{n-1}$ is known from the previous iteration. The (i,j,k) element of $\bar{R}^{n-1}$ is given by

$$R^{n-1} = Q - (Bh_{i-1}^{n-1} + Dh_{j-1}^{n-1} + Eh^{n-1} + Fh_{j+1}^{n-1} + Hh_{i+1}^{n-1} + Sh_{k+1}^{n-1} + Zh_{k-1}^{n-1}).$$

When $\xi_{max}^{n} < \varepsilon$, an arbitrary error criterion, a solution to equation 13 is obtained.

Derivation of the SIP algorithm requires 1) the relationships defining

$$\overline{\overline{(A+B)}} = \overline{\overline{LU}}; \qquad (14)$$

2) an appropriate definition of $\bar{\bar{B}}$; and 3) relationships among the elements of $\bar{\bar{A}}$ and $\overline{\overline{(A+B)}}$. The matrix $\overline{\overline{(A+B)}}$ is shown in figure 5 and the elements of $\overline{\overline{(A+B)}}$ or $\overline{\overline{LU}}$ are defined by

$$
\begin{aligned}
Z' &= a \\
A &= ae_{k-1} \\
T &= af_{k-1} \\
B' &= b \\
C &= e_{i-1}b \\
D' &= c \\
E' &= ag_{k-1} + bf_{i-1} + e_{j-1}\,c + d \\
F' &= d\,e \\
G &= f_{j-1}c \\
H' &= f\,d \\
U &= bg_{i-1} \\
W &= g_{j-1}c \\
S' &= gd \qquad (15)
\end{aligned}
$$

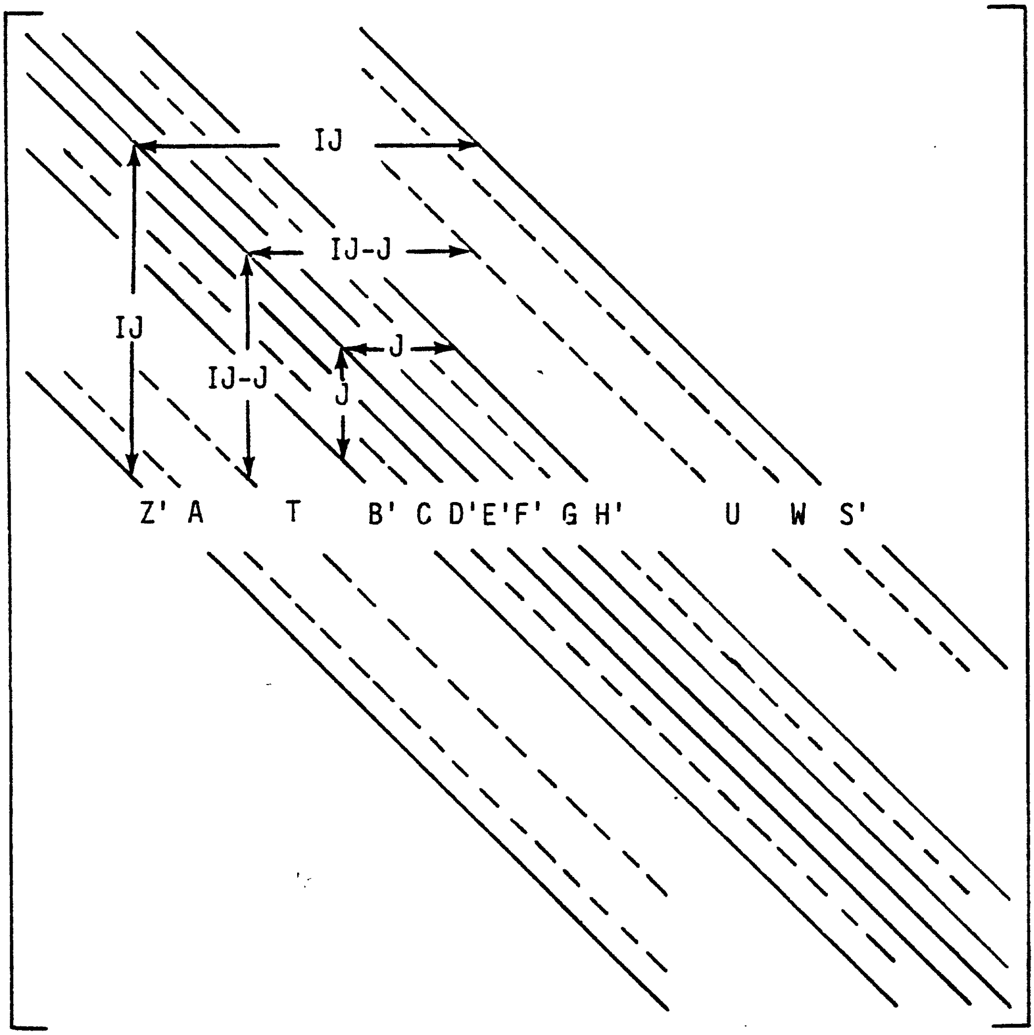

Figure 5.--Matrix $(\overline{\overline{A+B}})$.

The nodal values associated with the elements of $(\overline{\overline{A + B}})$ not in the original difference equation 9 (namely, those associated with the coefficients C,G,A,W,T,U) are shown in figure 6. Stone (1968) and Weinstein, Stone and Kwan (1969) have found it effective to define $\overline{\overline{B}}$ (there are many ways of defining $\overline{\overline{B}}$ that will satisfy equation 14) so that the contribution of these additional terms is partially cancelled. Their definition of $\overline{\overline{B}}$ can be seen by writing $\overline{\overline{B}}\overline{\xi}$ for node (i,j,k) as

$$
\begin{array}{lll}
C\ [\xi_{i-1,j+1} & -\ \omega & (-\xi + \xi_{i-1} + \xi_{j+1})] + \\
G\ [\xi_{i+1,j-1} & -\ \omega & (-\xi + \xi_{j-1} + \xi_{i+1})] + \\
A\ [\xi_{j+1,k-1} & -\ \omega & (-\xi + \xi_{j+1} + \xi_{k-1})] + \\
W\ [\xi_{j-1,k+1} & -\ \omega & (-\xi + \xi_{j-1} + \xi_{k+1})] + \\
T\ [\xi_{i+1,k-1} & -\ \omega & (-\xi + \xi_{i+1} + \xi_{k-1})] + \\
U\ [\xi_{i-1,k+1} & -\ \omega & (-\xi + \xi_{i-1} + \xi_{k+1})]
\end{array}
\qquad (16)
$$

in which ω is an iteration parameter. Weinstein, Stone and Kwan (1969) define different parameters for each of the three planes in space, but in practice use only one parameter each iteration. The term $(-\xi+\xi_{i-1} + \xi_{j+1})$ is the second-order correct approximation for $\xi_{i-1,j+1}$. The remaining elements of $\overline{\overline{B}}\overline{\xi}$ for node (i,j,k) in 16 have similar relationships. (See Remson and others, 1971, p. 226 for derivation of this type of approximation.)

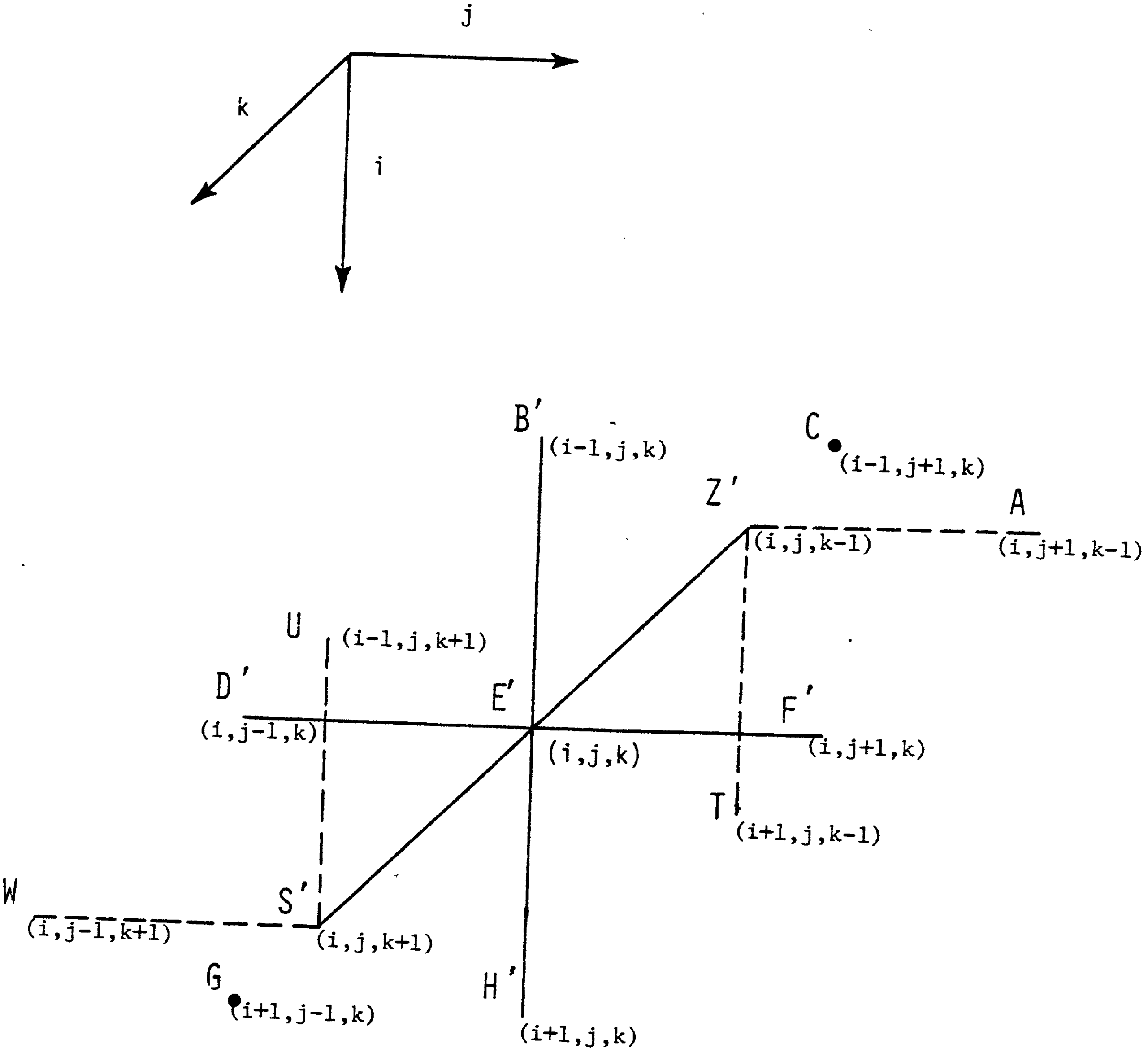

Figure 6.--Coefficients for normal ordering scheme.

With these definitions, equation 13 written for node (i,j,k) is

$$B\xi^n_{i-1} + D\xi^n_{j-1} + E\xi^n + F\xi^n_{j+1} + H\xi^n_{i+1} + S\xi^n_{k+1} + Z\xi^n_{k-1} +$$

$$\begin{aligned}
&C[\xi^n_{i-1,j+1} - \omega \ (-\xi^n+\xi^n_{i-1} + \xi^n_{j+1})] + \\
&G[\xi^n_{i+1,j-1} - \omega \ (-\xi^n+\xi^n_{j-1} + \xi^n_{i+1})] + \\
&A[\xi^n_{j+1,k-1} - \omega \ (-\xi^n+\xi^n_{j+1} + \xi^n_{k-1})] + \\
&W[\xi^n_{j-1,k+1} - \omega \ (-\xi^n+\xi^n_{j-1} + \xi^n_{k+1})] + \\
&T[\xi^n_{i+k,k-1} - \omega \ (-\xi^n+\xi^n_{i+1} + \xi^n_{k-1})] + \\
&U[\xi^n_{i-1,k+1} - \omega \ (-\xi^n+\xi^n_{i-1} + \xi^n_{k+1})] = R^{n-1}.
\end{aligned} \qquad (17\ )$$

By collecting coefficients associated with nodal positions in the original difference equation and using equations 15, the SIP algorithm can be derived as

$a = Z/(1+\omega\ (e_{k-1} + f_{k-1}))$

$b = B/(1+\omega\ (e_{i-1} + g_{i-1}))$

$c = D/(1+\omega\ (f_{j-1} + g_{j-1}))$

$A = ae_{k-1}$

$C = be_{i-1}$

$G = cf_{j-1}$

$W = cg_{j-1}$

$T = af_{k-1}$

$U = bg_{i-1}$

$$d = E + \omega(A + C + G + W + T + U) - ce_{j-1} - bf_{i-1} - ag_{k-1}$$

$$e = (F - \omega(A + C))/d$$

$$f = (H - \omega(G + T))/d$$

$$g = (S - \omega(W + U))/d. \qquad (18)$$

Equation 14 may be combined with equation 13 to give

$$\bar{\bar{L}}\bar{\bar{U}}\bar{\xi}^{n} = \bar{R}^{n-1}. \qquad (19)$$

Define the vector $\bar{V}^n$ by

$$\bar{\bar{U}}\bar{\xi}^{n} = \bar{V}^{n}. \qquad (20)$$

Then equation 19 becomes

$$\bar{\bar{L}}\bar{V}^{n} = \bar{R}^{n-1}. \qquad (21)$$

After solving equations 18, the intermediate vector $\bar{V}^n$ can be computed; for node (i,j,k) equation 21 is (refer to figure 3 for the elements of $\bar{\bar{L}}$)

$$aV^{n}_{k-1} + bV^{n}_{i-1} + cV^{n}_{j-1} + dV^{n} = R^{n-1}$$

or

$$V^{n} = (R^{n-1} - aV^{n}_{k-1} - bV^{n}_{i-1} - cV^{n}_{j-1})/d.$$

The vector $\bar{\xi}^n$ may then be computed by backward substitution.

Equation 20 for node (i,j,k) is (refer to figure 4 for the elements of $\bar{\bar{U}}$)

$$\xi^{n} + e\xi^{n}_{j+1} + f\xi^{n}_{i+1} + g\xi^{n}_{a+1} = V^{n}$$

or

$$\xi^{n} = V^{n} - (e\xi^{n}_{j+1} + f\xi^{n}_{i+1} + g\xi^{n}_{k+1}).$$

Stone (1968) and Weinstein and others (1969) recommend that a second ordering scheme be used every other iteration. In the second ordering scheme, j is swept through first in increasing order (j = 1,2,...,J), i is second in decreasing order (i = I, I-1, ...,1), and k last in decreasing order (k = K,'K'-1,...,1). (See figure 7.) This tends to give an overall symmetry to $(\overline{\overline{A+B}})$ for the two iterations (compare figures 6 and 8) and speeds convergence.

The algorithm for the second (or 'reverse') ordering scheme is derived in a manner analogous to that for the 'normal' ordering scheme given above. The SIP reverse algorithm is outlined below:

$$
\begin{aligned}
a &= S/(1+\omega(e_{k+1} + f_{k+1})) \\
b &= H/(1+\omega(e_{i+1} + g_{i+1})) \\
c &= D/(1+\omega(f_{j-1} + g_{j-1})) \\
A &= ae_{k+1} \\
C &= be_{i+1} \\
G &= cf_{j-1} \\
W &= cg_{j-1} \\
T &= af_{k+1} \\
U &= bg_{i+1} \\
d &= E + \omega\,(C+G+A+W+T+U) - ag_{k+1} - bf_{i+1} - ce_{j-1} \\
e &= (F-\omega(C+A))/d \\
f &= (B-\omega(G+T))/d \\
g &= (Z-\omega(W+U))/d. \qquad (22)
\end{aligned}
$$

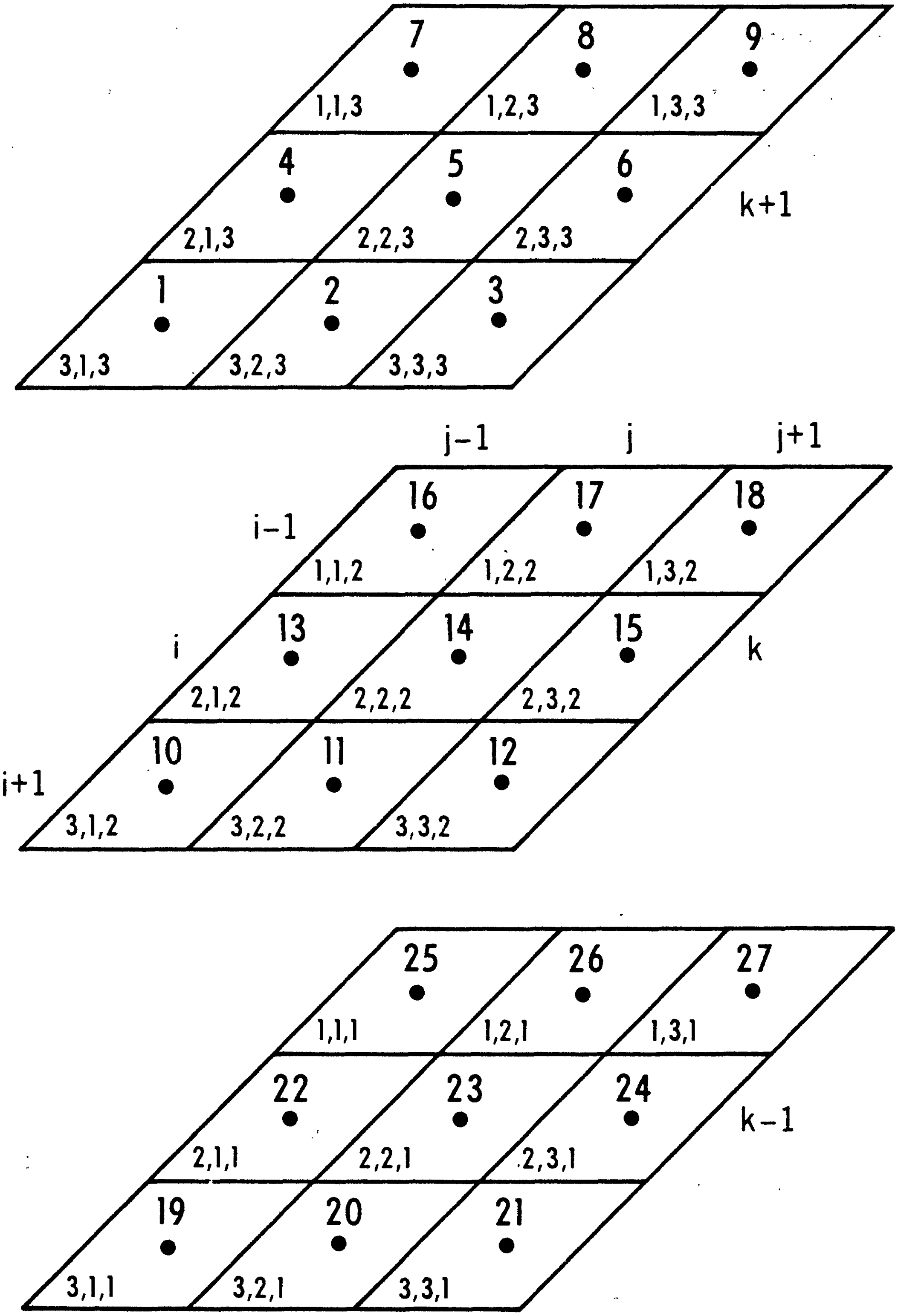

Figure 7.--Reverse ordering of nodes for a 3 X 3 X 3 problem. The numerical values of the indices (i,j,k) for this problem are given in the lower left-hand corner of each block.

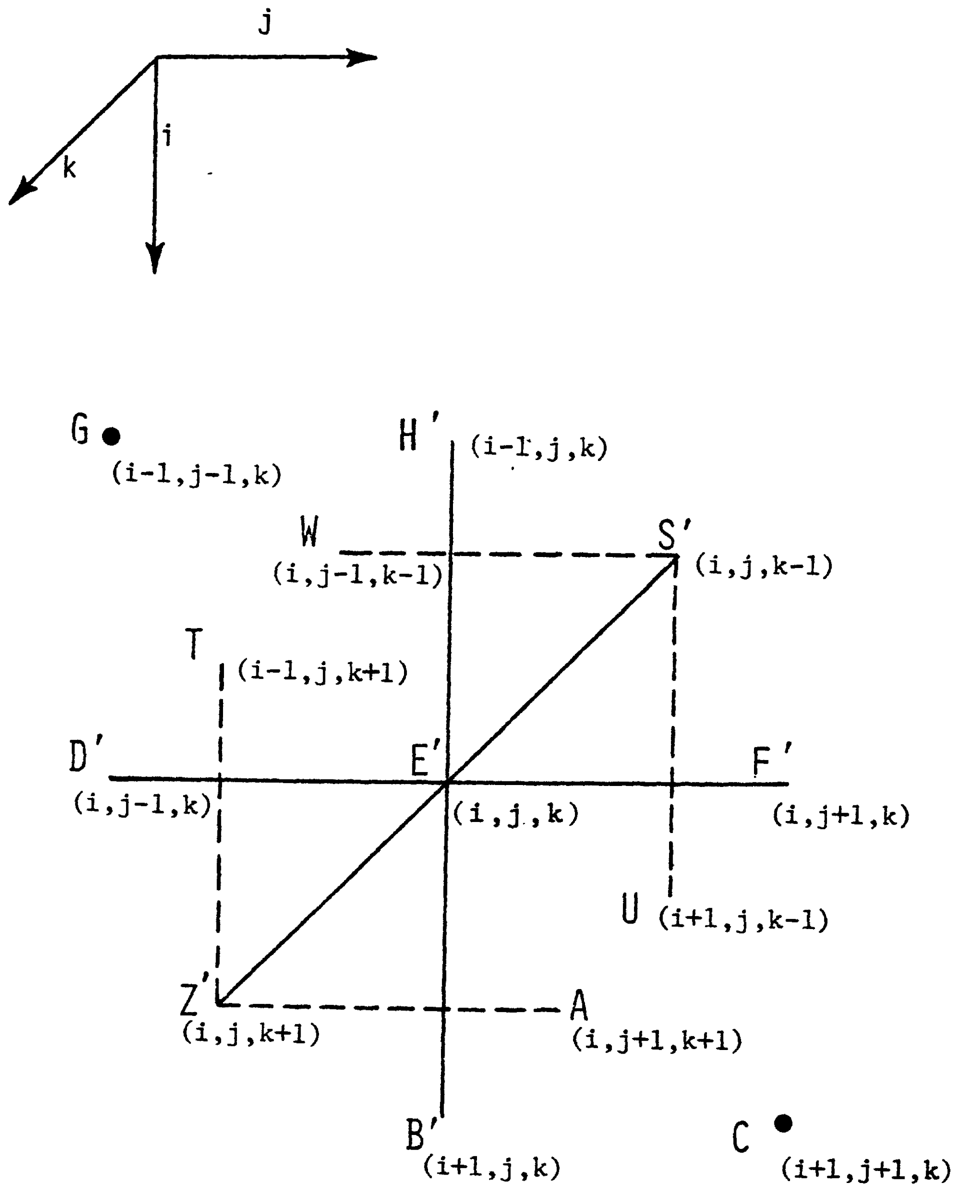

Figure 8.--Coefficients of reverse ordering scheme.

For node (i,j,k) equation 21 is

$$aV^n_{k+1} + bV^n_{i+1} + cV^n_{j-1} + dV^n = R^{n-1}$$

or

$$V^n = (R^{n-1} - aV^n_{k+1} - bV^n_{i+1} - cV^n_{j-1}) /d.$$

The vector $\bar{\xi}^n$ is computed with equation 20 which, for node (i,j,k), is

$$\xi^n + e\xi^n_{j+1} + f\xi^n_{i-1} + g\xi^n_{k-1} = V^n$$

or

$$\xi^n = V^n - e\xi^n_{j+1} - f\xi^n_{i-1} - g\xi^n_{k-1}.$$

Iteration parameters

A sequence of iteration parameters ranging from 0 to 1 is cycled until convergence is achieved. The minimum parameter is not critical and 0 is arbitrarily chosen. The maximum parameter is given by

$$1-\omega_{max} = \underset{\text{(over grid)}}{\text{Min}} \left[\frac{\pi^2}{2J^2(1+\rho_1)}, \frac{\pi^2}{2I^2(1+\rho_2)}, \frac{\pi^2}{2K^2(1+\rho_3)}\right] \qquad (23a)$$

in which

$$\rho_1 = \frac{K_{yy[i,j,k]}(\Delta x_j)^2}{K_{xx[i,j,k]}(\Delta y_i)^2} + \frac{K_{zz[i,j,k]}(\Delta x_j)^2}{K_{xx[i,j,k]}(\Delta z_k)^2} \qquad (23b)$$

$$\rho_2 = \frac{K_{xx[i,j,k]}(\Delta y_i)^2}{K_{yy[i,j,k]}(\Delta x_j)^2} + \frac{K_{zz[i,j,k]}(\Delta y_i)^2}{K_{yy[i,j,k]}(\Delta z_k)^2} \qquad (23c)$$

$$\rho_3 = \frac{K_{xx[i,j,k]}(\Delta z_k)^2}{K_{zz[i,j,k]}(\Delta x_j)^2} + \frac{K_{yy[i,j,k]}(\Delta z_k)^2}{K_{zz[i,j,k]}(\Delta y_i)^2} \qquad (23d)$$

Equations 23 are the same ones used to calculate the minimum parameter for the iterative alternating-direction implicit procedure (ADI) and are based on a von Neumann error analysis of the normalized ADI equations. (See Weinstein and others, 1969, for references.) Commonly 4 to 10 parameters are used in one cycle and are arranged geometrically between minimum and maximum with the equation

$$\omega_{\ell+1} = 1+(1-\omega_{\max})^{\ell/L-1}, \quad \ell = 0,1,\ldots,L-1 \qquad (24)$$

in which ℓ is the iteration parameter index;

L is the number of parameters in a cycle.

In the model, equation 23b, 23c, and 23d are modified to use B,D,F,H,S and Z defined by equations 8. For example, equation 23b is computed as

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

The sequence of parameters computed with equations 23 and 24 gives rapid convergence for most anisotropic problems ($K_{xx} \simeq K_{yy} >> K_{zz}$). For problems with isotropic layers, nowever, the parameters, computed with these equations may give slower convergence. Based on an analysis of the contribution of terms in equations 23, the second terms of equations 23 were dropped and the resulting sequence of parameters generally give good convergence rates for both isotropic and anisotropic simulations. However, if the sequence of parameters computed by the equations in the program are all (except the first parameter) close to 1.0 and if this results in slow convergence or even divergence, bypass the computations in the model and insert WMAX ≃ 0.99863. The resulting sequence of parameters (such as 0.0, 0.80772, 0.96303, 0.99289, 0.99863) may give a satisfactory rate of convergence.

Boundaries in the numerical scheme

Within the limits of the finite-difference grid the irregular geometry of the porous medium is simulated by assigning zero transmissivity to blocks outside the system. These blocks, however, must be included in the SIP algorithm in order to preserve the required seven-diagonal coefficient matrix $\bar{\bar{A}}$. Constant-head boundaries are treated without using special conditional statements by skipping constant-head nodes in the algorithm.

PRACTICAL CONSIDERATIONS FOR APPLICATION

Initial Conditions

In many simulations, the important results are not the computed head, but the changes in head caused by a stress, such as pumping of wells. For this objective, if the flow equations for every layer are linear (that is, all hydraulic units are confined), the computed drawdown can be superimposed on the natural flow system. Consequently, the initial head for the simulation may be horizontal and there is no need to impose a natural flow system as the initial condition. However, if the user wishes to start a simulation using the head distribution in the natural flow system as the initial surface, a steady state simulation can be made with the model to compute the initial surface. This type of simulation is also useful to check the transmissivity and leakance distribution in the model.

If initial conditions are specified so that transient flow is occurring in the system at the start of the simulation, it should be recognized that the water levels will change during the simulation, not only in response to the pumping stress, but also due to the initial conditions. Initial conditions of this type are appropriate if the simulation is started part way through the history of development of the aquifer system.

Treatment of boundary conditions

Boundaries that can be treated by the model are of two types: constant head and constant flux. Constant-head boundaries are specified by assigning a negative storage coefficient to the nodes that define the constant-head boundary. A constant flux may be zero (impermeable boundaries) or have a finite value. A zero-flux boundary is treated by assigning a zero transmissivity to nodes outside the boundary. The harmonic mean of the transmissivity at the cell boundary is zero, and consequently, the flux across the boundary is zero. A no-flow boundary is inserted around the border of each layer of the model as a computational expediency, and constant head or finite-flux boundaries are placed inside this border. A finite-flux boundary is treated by assigning recharge wells to the appropriate nodes.

Treatment of Confining Layers

A confining bed in which storage is insignificant may be simulated in either of two ways: 1) It may be represented by one or more layers of nodes as illustrated in figure 9a with TK(I,J,K) computed in the program by equation 26c (see Appendix II). This approach is necessary if horizontal components of flow in the confining bed are significant 2) If horizontal components of flow in the confining bed can be ignored, the system can be simulated in the same manner as the quasi three-dimensional model described by Bredehoeft and Pinder (1970) with a savings in computer time and storage. For this approach the confining beds are not represented by layers of nodes (figure 9b). Instead, the effects of vertical flow through the confining bed are incorporated in the vertical components of hydraulic conductivity of the adjacent aquifers. In practice, this is accomplished by reading the TK values, which are normally computed by equation 26c, with the rest of the data input.(K'-1 sets of TK values must be read). The TK values are equal to the ratio K_{zz}/b for each confining bed and $\Delta z = 1.0$ for all layers. This approach is designed to be used in solving equation 4. However, for certain simple problems in which the confining bed is horizontal, TK values can be read and nodes representing the confining bed eliminated when the model is used to solve equation 3 if the appropriate Δz values are retained in the data input.

If storage in the confining bed is significant, a number of layers of nodes will be required to give a good approximation to the gradients near the boundaries of the confining layer. To reduce computation time and storage

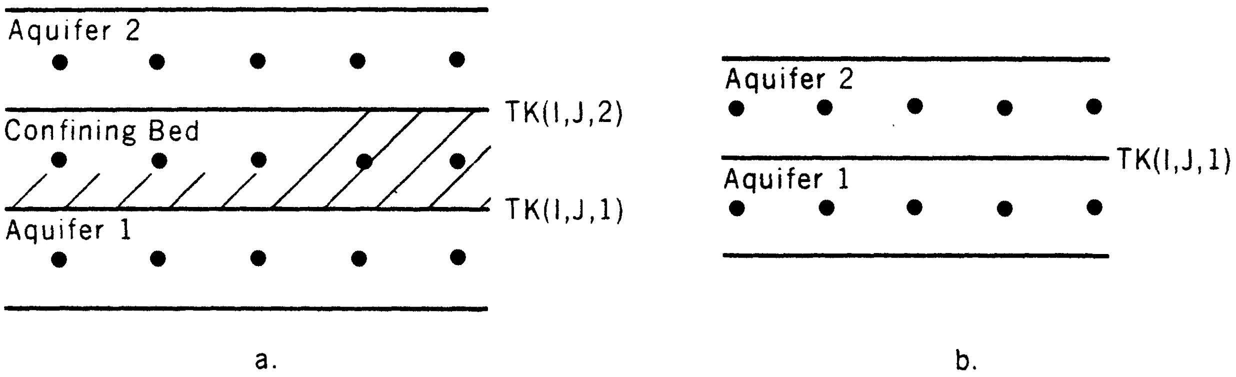

Figure 9.--Simulating two aquifers separated by a confining bed with three layers of nodes (a) or two layers of nodes (b).

requirements, an analytic approximation analogous to that used in the simulator for flow in two dimensions (Bredehoeft and Pinder, 1970, P.C. Trescott, G.F. Pinder and S.P. Larson, written communication, 1975) may be more suitable.

Designing the finite-difference grid

In designing a finite-difference grid, the following considerations should be kept in mind:

1. Nodes representing pumping and observation wells should be close to their relative positions to facilitate calibration. If several pumping wells are close together, their discharge may be lumped, and assigned to one node since discharge is distributed over the volume of the block.

2. Boundaries within the project area should be located accurately. Distant boundaries can be located approximately and with fewer nodes by expanding the grid. In expanding a finite-difference grid in the positive J direction, experience has shown that restricting the ratio $\Delta x_j/\Delta x_{j-1} \leq 1.5$ will avoid large truncation errors and possible convergence problems. This rule applies in all three dimensions. If the quasi three-dimensional approach is used and TK values are used with the data input, set $\Delta z = 1.0$ for all layers.

3. Place nodes closer together in areas of rapidly changing transmissivity in each layer.

4. The grid should be oriented so that a minimum of nodes are outside the porous medium. Orienting the grid with respect to latitude and longitude or some other geographic grid system is a secondary consideration. However, if the aquifer is anisotropic, orient the grid with its axes parallel to the

principal directions of the transmissivity tensor. Otherwise, the flow equation would include cross-product terms which are not considered in the SIP algorithm.

5. Number the rows in the short dimension for plotting maps on the line printer or for plotting data with an X-Y drum plotter. On these plots, the X direction is vertical and, for practical purposes, this dimension is unlimited. The Y direction is across the page which limits this dimension to the maximum width of the page.(See figure 10).

6. The core requirements and computation time are proportional to the number of nodes representing the porous medium.

REFERENCES

Bredehoeft, J.D., and Pinder, G.F., 1970, Digital analysis of areal flow in multiaquifer ground water systems: a quasi three-dimensional model: Water Resources Res., v. 6, no. 3, p. 883-888.

Cooper, H.H. Jr., 1966, The equation of groundwater flow in fixed and deforming coordinates: Jour. Geophysical Res., v. 71, no. 20, p. 4785-4790.

Jacob, C.E., 1950, Flow of ground water: Chapter V, p. 321-386, in Engineering Hydraulics, edited by Rouse: New York, John Wiley and Sons.

Remson, I., Hornberger, G.M., and Molz, F.J., 1971, Numerical Methods in Subsurface Hydrology: New York, Wiley-Interscience, 389 p.

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

Weeks, J.B., Leavesley, G.H., Welder, F.A., and Saulnier, G.J., 1974, Simulated effects of oil-shale development on the hydrology of Piceance Basin, Colorado: U.S. Geol. Survey Prof. Paper 908, 84 p.

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

APPENDIX I

NOTATION

a,b,c,d	elements of lower triangular factor $\bar{\bar{L}}$;
$A,B',C,D',E',F',G,$ H',S',T,U,W,Z'	coefficients of $(\overline{\overline{A+B}})$;
$\bar{\bar{A}}$	coefficient matrix;
b	saturated thickness of a hydraulic unit (L);
B,D,E,F,H,S,Z	coefficients in difference equation;
$\bar{\bar{B}}$	modifying matrix;
e,f,g	elements of upper triangular factor, $\bar{\bar{U}}$;
h	hydraulic head (L);
$\hat{h}$	hydraulic head at the previous time step (L);
i	index in the y dimension;
I	number of rows;
j	index in the x dimension;
J	number of columns;
k	index in the z dimension;
K'	number of layers;
K	hydraulic conductivity (LT^{-1});
K_{xx},K_{yy},K_{zz}	principal components of the hydraulic conductivity tensor (LT^{-1});
ℓ	iteration parameter index;
L	number of parameters in a cycle;
$\bar{\bar{L}}$	lower triangular factor of $(\overline{\overline{A+B}})$;
n	iteration index;
N_a	number of arrays required for the options;

q_{re}	volumetrix flux per unit area of the uppermost hydraulic unit (LT^{-1});
Q	known term in difference equation;
Q_w	well discharge (L^3T^{-1});
R^{n-1}	residual for previous iteration;
S'	storage coefficient (dimensionless);
S_s	specific storage (L^{-1});
T_{xx}, T_{yy}	principal components of the transmissivity tensor (L^2T^{-1});
$\overline{\overline{U}}$	upper triangular factor of $(\overline{\overline{A+B}})$;
$\overline{V}$	intermediate vector in SIP algorithm;
W	volumetric flux per unit volume (T^{-1});
Δt	time increment (T);
Δx	space increment in the x direction (L);
Δy	space increment in the y direction (L);
Δz	space increment in the z direction (L);
ε	closure criterion;
$\overline{\xi}$	vector of change in head for an iteration;
ρ_1, ρ_2, ρ_3	hydraulic conductivity ratios;
ω	iteration parameter.

APPENDIX II

COMPUTER PROGRAM

Main Program

The main program first assigns storage space to arrays required in the simulation. Storage space is reserved in a vector Y and is allocated to the arrays based on the dimensions of the problem specified on card 3 of the data deck. (See Appendix III). The minimum dimension of Y is approximately

$$YDIM \simeq 15\ IJK' + N_a\ IJ. \tag{25}$$

in which N_a is the sum of arrays required for options (see table 1).

Table 1.--Additional arrays required for options

Option	Number of arrays
Water-table conditions in upper unit	2
Recharge from precipitation	1

Equation 25 is approximate, but normally will give a value that is sufficient for the simulation. The exact dimension required is printed on the first page of the output as "WORDS OF VECTOR Y USED = XXXX". For new simululations, the program does not need to be recompiled as long as the dimension of Y is large enough and FORTRAN statements do not need to be modified.

After allocation of storage space to the arrays, the addresses of the first element of each array are passed to the subroutines. (See table 2 for details). In table 2 the variables giving the dimensions of the arrays are defined in Appendix V; the first array is the only double precision array.

The remainder of the main program controls the sequence of computations.

Subroutine DATAI

Instructions for the preparation of the data deck are given in Appendix III. Data may be input to the model in any consistent set of units in which second is the time unit. It is organized into four groups: Data in groups I and II

Table 2.--Arrays passed to the subroutines and their relative location in the vector Y.

Array	Sequence number in vector Y	Subroutine						Dimensions
		DATAI	STEP	SOLVEI	COEF	CHECKI	PRNTAI	
PHI	1	X	X	X	X	X	X	(IO, JO, KO)*8
STRT	2	X	X	X	X	X	X	IO, JO, KO
OLD	3	X	X	X	X	X		
T	4	X	X	X	X	X	X	
S	5	X	X	X	X	X	X	
TR	6	X	X	X	X	X		
TC	7	X	X	X	X	X		
TK	8	X	X	X	X	X		IK, JK, K5
WELL	9	X	X	X	X	X	X	IO, JO, KO
EL	10			X				
FL	11			X				
GL	12			X				
V	13			X				
XI	14			X				
DELX	15	X	X	X	X	X	X	JO
DELY	16	X	X	X	X	X	X	IO
DELZ	17	X	X	X	X	X		KO
DDN	18		X					IMAX
FACT	19	X	X	X	X	X		KO,3
TEST3	20		X	X				ITMX1
JFLO	21					X		NCH, 3
FLOW	22					X		NCH
PERM	23	X			X			IP, JP
BOTTOM	24	X			X			
QRE	25	X			X	X		IQ, JQ

are the simulation options and scalar parameters; group III cards are used to initialize the arrays. These three groups are required for each new simulation. Group IV contains data that varies with each new pumping period. The program permits changing well discharge and the time parameters each pumping period, but the program can be modified to read other data with this set of cards.

If the simulation is a continuation of a previous run, the initial head for the continuation (not the starting head for the run) is read either from cards or from disk depending on how the results of the previous run were saved. (See the section on technical information for details of the data set on disk.) A three-dimensional simulation may require a large number of data cards. To reduce the number of cards that must be read with each run, the program includes options to place the arrays on disk and, on subsequent runs, read the data from disk rather than from cards. The data requirements for this option are included in Appendix III and definition of the direct access file on disk is explained in the section on technical information.

Time Parameters. The time parameters include the initial time step, DELT; a multiplication factor for increasing the size of the time step, CDLT; the number of time steps; NUMT; and the simulation period, TMAX. Since the rate of water-level change decreases during a pumping period, the time step is increased by the factor CDLT each step (commonly 1.5). For any time step (k) the time increment is given by

$$\mathrm{DELT}_k = \mathrm{CDLT} * \mathrm{DELT}_{k-1}.$$

DELT_o is the time step recorded on the data card.

The program has two options for selecting the time parameters:

1. To simulate a given period of time, select CDLT and an appropriate DELT_o, and set NUMT greater than the expected number of time steps. The program computes the required initial DELT_o (which will not exceed the value of DELT_o coded on card 1 of group IV) and NUMT to arrive exactly at TMAX on the final step.

2. To simulate a given number of time steps, set TMAX greater than the expected simulation period and the program will use $DELT_o$, CDLT and NUMT as specified on the time parameter card.

To minimize the error due to approximation of the time derivative, several time steps should be simulated before the first step at which results are displayed. This suggestion should be followed unless the system is nearly at steady state before the results are needed. In this case a one-step simulation may be satisfactory but this approach should be checked by making one run as a multi-step simulation so that the results can be compared.

For steady-state simulations, set the storage coefficient of each layer to zero. Compute for one time step of any length (for example, set TMAX = 1, NUMT = 1, CDLT = 1, DELT = 24) and the program should iterate to a solution. The maximum permitted number of iterations (ITMAX) should be larger for steady-state than for transient simulations.

Initialization. In addition to reading data and computing the time parameters this routine initializes other arrays and scalar parameters. In particular, note that the division of well discharge by the area of the block needs to be done only once for each pumping period.

Subroutine STEP

Subroutine STEP initializes variables for a new time step and controls the printing and punching of results for designated time steps. In this routine, the computed head values, cumulative time and cumulative values for the mass balance parameters are punched on cards if PUN2 is specified in the options in group I of data input or are written on a previously defined data set on disk if DK2 is specified in the options. (See the section on technical information for details about the data set on disk).

Subroutine SOLVE

This subroutine computes the SIP iteration parameters and has the SIP normal and reverse algorithms which are outlined in the section on numerical solution. The algorithm is shortened for computation on the lowermost and uppermost layers.

In this routine the usual (I,J,K) notation has been replaced in favor of single-subscript notation. Less time is involved in finding the value of a variable with a single subscript than in finding the value of one with three subscripts and as a consequence computational efficiency is improved. The subscripts used in this notation are defined in the code.

Computational efficiency is also improved by including the computation of the term $S'/\Delta t$ and the coefficients B,D,F,H,S, and Z in this routine rather than in COEF.

If the permitted number of iterations for the time step is exceeded, the message "EXCEEDED PERMITTED NUMBER OF ITERATIONS" is printed. Following the message, the mass balance, head matrix and other data specified in the options are printed. This information is useful in determining the cause of non convergence. The mass-balance parameter values and head values for the last iteration are punched if PUNC is specified in the option or written on disk if DK2 is specified. These results can be used to extend the simulation if it appears that a solution can be obtained.

Subroutine COEF

If the upper hydraulic unit is under water-table conditions, its transmissivity as a function of saturated thickness is recomputed every iteration. For those nodes where the computed head drops below the bottom elevation of the unit, a message 'NODE I, J IN LAYER K GOES DRY' is printed. A similar message is printed if a well node goes dry. The hydraulic conductivity at these nodes is set to zero and computation continues. No provision is made to permit these blocks to resaturate because the additional code for this special situation is not warranted in a general program.

The T coefficients TR, TC and TK may be computed once and saved for artesian units; for water-table units, TR and TC are recomputed every iteration and TK stays constant until block K+1 desaturates, then TK(I,J,K) = 0.0. They are defined as

$$\mathrm{TR}(I,J,K) = \frac{2T_{xx[j+1]}\,T_{xx}}{T_{xx}\Delta x_{j+1} + T_{xx[j+1]}\Delta x}\,; \qquad (26a)$$

$$\mathrm{TC}(I,J,K) = \frac{2T_{yy[i+1]}\,T_{yy}}{T_{yy}\Delta y_{i+1} + T_{yy[i+1]}\Delta y}\,; \qquad (26b)$$

$$\mathrm{TK}(I,J,K) = \frac{2(bK_{zz})_{k+1}(bK_{zz})}{(bK_{zz})\Delta z_{k+1} + (bK_{zz})_{k+1}\Delta z}. \qquad (26c)$$

Subroutine CHECK

A mass balance is computed in this routine. The results are expressed in two ways: 1) as a cumulative volume of water from each source and each type of discharge, and 2) as rates for the current time step. Note that 'leakage' and 'evapotranspiration' appear in the mass balance but are set to zero because these options are not included in the current version of the model.

In the cumulative mass balance, storage is treated as a source of water. Flow to and from constant-head boundaries is computed with Darcy's law using the gradients from constant-head nodes to adjacent nodes inside the porous medium. Other computations in the algorithm are self-explanatory. The difference between the sum of recharge sources and sum of discharges from the system is usually less than one percent.

To the right of the cumulative mass balance are printed the flow rates for the current time step. Below the cumulative mass balance are printed the flow rates to individual constant head nodes (if included in the simulation). This is followed by the net flow rate to the top layer from below and the net flow rate to the bottom layer from above.

The last item is the listing of the absolute value of the maximum head change for each iteration. This information is useful if convergence is slow with SIP because it may indicate that a slightly larger error criterion will give a satisfactory solution with considerably fewer iterations.

Subroutine PRNTAI

This routine prints maps of drawdown and(or) hydraulic head for selected layers. Up to three characters are plotted for each block with the rightmost character as close to the location of the node as the printer will allow. The user specifies XSCALE and YSCALE, the multiplication factors required to change from units used in the model to units used in the map (in general they should be the same); DINCH, the number of map units per inch; FACT1 and FACT2, the multiplication factors for adjusting the values of drawdown and head to be plotted, respectively; MESUR, the name of the unit used on the map, and vectors LEVEL1 and LEVEL2 which contain the layers for which maps are to be printed. As an example, assume that the length unit used in the model is feet, the map is to be scaled at 3 miles per inch and drawdown values to the nearest foot and head values to the nearest 10 feet are to be plotted for layers 1 and 3. Then XSCALE = YSCALE = 5280, DINCH = 3, FACT1 = 1; FACT2 = 0.1; MESUR = MILES; LEVEL1 (1) = LEVEL2 (1) = 1; and LEVEL1(2) = LEVEL2(2) = 3.

To print a map of maximum possible size, number the rows in the short dimension to take advantage of the orientation of the map on the computer page where the X direction is vertical and the Y direction horizontal. (See figure 10). The origin is the upper left-hand corner of the block for row 2, column 2. Orienting the map with the origin in the upper left hand corner, the right and bottom sides of the map include the node locations for the second to last column and row, respectively. The border is located to the nearest inch outside these node locations and may or may not fall on the block boundaries depending on the scaling. The map is automatically centered on the page and is limited to a maximum of 12 inches (30 cm) in the Y direction. If the parameters for a map are specified such that the Y dimension is more than 12 inches (30 cm) adjustments are automatically made to fit the map within this limit. A common mistake is to specify a value for Y scale that is less than 1.0. This generates the message 'NOTE: GENERALLY SCALE SHOULD BE > OR = 1.0' and a suitable adjustment is made to DINCH. In the X direction the map is limited only by the dimension of the NX vector (for example, when the dimension of NX is 100, the map is limited in the X direction to 100 - 1 = 99 inches [250 cm]). Several parameters (PRNT, BLANK, N1, N2, N3, and XN1) are initialized in the BLOCK DATA routine to values that assume the line printer

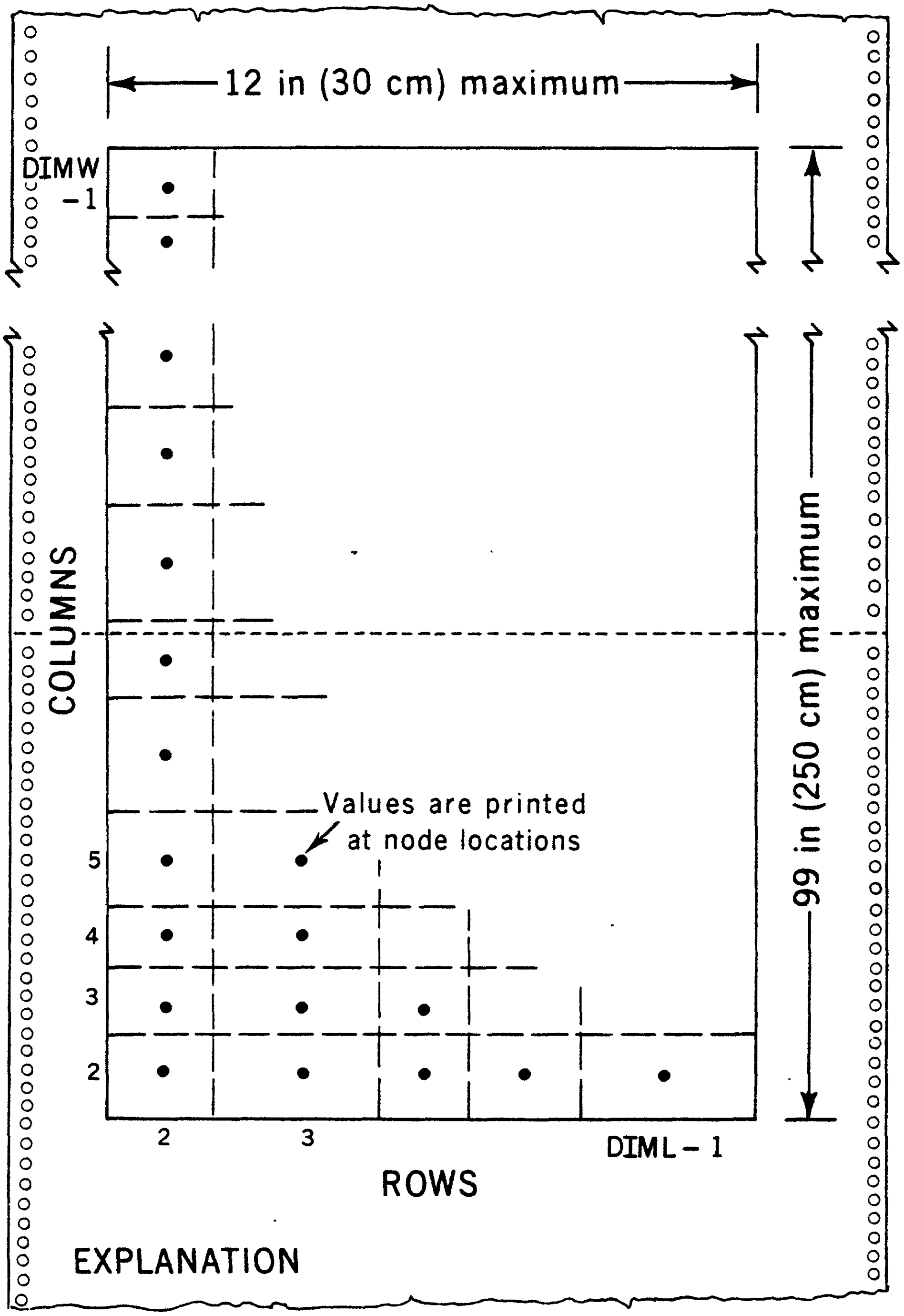

Figure 10.--Orientation of maps on computer page.

prints 6 lines per inch, 10 characters per inch, and 132 characters per line. These parameter values may need to be changed for a line printer with other specifications.

The PRNTAI subroutine can be modified to cycle a set of alphameric symbols for drawdown. If this type of map is desired, remove the C from column 1 of statements PRN1060 and PRN1230. This will cycle the symbols 1,2,3,4,5,6,7,8,9,0 for drawdown. To plot a different set of symbols will require modification of the initialization of SYM in BLOCK DATA. To cycle more than 10 symbols will require more extensive changes to the initialization of SYM and modifications to the code in ENTRY PRNTA.

BLOCK DATA

The BLOCK DATA routine initializes scalar parameters and arrays used in PRNTAI and other subroutines.

Technical Information

Use of disk facilities for storage of array data and interim results. Options are included in the program to enable storage and retrieval of array data (STRT, S, T, etc.) and the saving of interim head values without punching them on cards. Use of these options can be particularly beneficial at remote terminals with low speed data transmission or without punch output capability. Also, the type of read statements used afford more efficient data transmission from disk than from cards.

Storage of array data is accomplished via a direct access data set that is defined by a DEFINE FILE statement in the main program (card MAN0310) and by a DD statement in the JCL string used to execute the program. To establish the data set, the DEFINE FILE statement and the DD statement must indicate the amount of space that is required. The DEFINE FILE statement takes the following form:

```
DEFINE FILE 2(XX,???,U,KKK)                                      MAN0310
```

where *???* is the number of nodes per layer for the problem being solved (IO * JO) and XX is the number of records to be reserved for data set storage. The value of XX is 3 times KO, plus KO-1 if TK values are assigned, plus 2 for the water table option, plus 1 for the recharge option. Parameters U and KKK are indicators and do not vary.

The DD statement contains information, such as account number, that will be different for each user. Also, the first reference to the data set is somewhat different from subsequent references. To utilize one of the disk packs for semi-permanent storage of user data, the first reference to the data set will take the following general form if the FORTGCG procedure is used to compile and execute the program.

```
//GO.FT02F001  DD  DSN=Data Set Name,
//   UNIT=ONLINE,DISP=(NEW,KEEP),
//   SPACE=(????,(XX)),DCB=(RECFM=F)
```

where *????* is the number of bytes per record that are to be reserved and should be set equal to IO * JO [*4].

When this initial allocation is processed the system will indicate in the HASP system log, JCL string output, the volume on which the data set was established (for example, SYS011 or SYS015). Subsequent use of the data set must indicate this information by modifying the underlined parameters in the initial reference to the data set. Thus the DD statement will read:

```
//GO.FT02F001  DD  DSN=Data Set Name,
//  UNIT=ONLINE,DISP=SHR,VOL=SER=YYYYYY
```

where the DSN parameter is the same as the initial run and YYYYYY indicates the volume (for example, SYS015) on which the data set was established by the initial run. The individual data arrays that are to be stored and later retrieved from this data set are specified on the parameter card for each array. These specifications are discussed completely in Appendix III.

Interim results. The initial head, cumulative simulation time, and mass balance parameters may be read from a data set on disk if DK1 is specified in the options. These data are the results of a previous simulation run and were written on disk because DK2 was specified in the previous run. They permit intermediate results to be examined before extending the simulation period. The same procedure also applies in simulations where the permitted number of iterations is exceeded.

The unit number for the data set is 4; an example of the JCL required to generate space for the data set on a disk is given below:

```
//STEP1 EXEC PGM=IEFBR14
//SYSUT1  DD  DSN=Data Name Set, VOL=SER=Disk Name,UNIT=3330,DISP=(NEW,KEEP),
//   SPACE=(TRK,(5,1)),DCB=(RECFM=VBS,BLKSIZE=13030)
```

Note that with the data set defined this way an extra 28K bytes of storage are required for the two buffers. If the block size is reduced, the storage required for the buffers is reduced correspondingly. If a procedure such as FORTGCG is used, insert a card analogous to

```
//GO.FT04F001  DD  UNIT=3330,VOL=Disk Name,DISP=SHR,DSN=Data Set Name
```

before the `//GO.SYSIN  DD  *` card to define unit 4.

JCL to execute a compiled program which includes unit 4 follows:

```
//Job card
//A   EXEC  PGM= Program Name, REGION=410K,TIME=3
//STEPLIB  DD  DSN='Data Set Name',VOL=SER=Disk Name,UNIT=3330,DISP=SHR
//FT04F001 DD  UNIT=3330,VOL=SER=Disk Name, DISP=SHR,DSN=Data Set Name
//FT06F001 DD  SYSOUT=A
//FT07F001 DD  SYSOUT=B
//FT05F001 DD *

DATA
/*
```

Unless the program is modified, writing on unit 4 destroys data previously written on this unit.

Storage Requirements. The amount of core that is needed can be found in table 3.

Table 3--Core requirements for compiling and executing program

Item	K bytes required
For compilation	
FORTRAN G, LEVEL 21	120
FORTRAN H, OPT=2	250
For execution	
FORTRAN G, LEVEL 21	88
FORTRAN H, OPT=2	72
BUFFERS	
Unit 4	28
Unit 2	$((IO*JO*4+Z^{\dagger})/1000)*2$
Vector Y	$X^{\dagger\dagger}/256$

[†] Round IO*JO*4 up to the nearest 2000 bytes (for example, if IO*JO*4=6400, let Z=1600 and the buffer space for Unit 2 will be 16K).

[††] X is the dimension of Y in the main program.

Computation time. Computation time is a function of many variables but as an approximation, about 0.001 second of CPU time on the IBM 370/155 is required for each interior node each iteration using the FORTRAN H, OPT = 2 compiler. For example, a steady-state problem designed by Weeks and others (1974) with 2116 interior nodes requiring 25 iterations to satisfy the error criterion took 51 seconds of CPU time.

Using a source code generated with the FORTRAN G, LEVEL 21 compiler, the computation time is increased by about one half.

FORTRAN IV. The program includes several FORTRAN IV features that are not in ANS FORTRAN (for example, ENTRY, END parameter in read statement, mixed-mode expressions, G format code, literal enclosed in apostrophes). If the program is used at a computer center where the FORTRAN compiler does not include these extensions, programmers at the selected installations may be available to modify the computer code as necessary.

Program Limitations

The model documented in this report was motivated primarily by a need to find a substitute for Bredehoeft and Pinder's (1970) quasi three-dimensional model. Consequently, the model should be reasonably free of errors for this type of simulation. The model has not been tested on all types of simulations in which equation 3 is being solved. Consequently, undiscovered errors in the logic may appear as the model is applied to new problems. I would appreciate hearing about 'bugs' in the logic so that corrections can be made and other users informed.

The finite-difference model for simulation of ground-water flow in two dimensions (P.C. Trescott, G.F. Pinder and S.P. Larson, written communication, 1975) includes evapotranspiration and an approximation to transient effects of confining beds in the source term, and permits an aquifer to change from artesian to water-table conditions. Some of these features (particularly evapotranspiration) can be added to the three-dimensional model with only moderate changes to the code.

APPENDIX III

DATA DECK INSTRUCTIONS

Group I: Title, Simulation Options and Problem Dimensions

This group of cards, which are read by the main program, contains data required to dimension the model. To specify an option on card 4 punch the characters underlined in the definition. For an option not used, that section of the card 4 can be left blank.

Note: Default typing of variables applies for all data input.

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
1	1-80	20A4	HEADING	Any title the user wishes to print on one line at the start of output.
2	1-52	I3A4	"	
3	1-10	I10	IO	Number of rows
	11-20	I10	JO	Number of columns
	21-30	I10	KO	Number of layers
	31-40	I10	ITMAX	Maximum number of iterations per time step
	41-50	I10	NCH	Number of constant head nodes
4	1-4	A4	IDRAW	DRAW to print drawdown
	6-9	A4	IHEAD	HEAD to print hydraulic head
	11-14	A4	IFLOW	MASS to compute a mass balance
	16-18	A3	IDK1	DK1 to read initial head, elapsed time, and mass balance parameters from unit 4 on disk
	21-23	A3	IDK2	DK2 to write computed head, elapsed time, and mass balance parameters on unit 4 (disk)
	26-29	A4	IWATER	WATE if the upper hydrologic unit is unconfined
	31-34	A4	IQRE	RECH for a constant recharge that may be a function of space

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
4	36-39	A4	IPU1	PUN1 to read initial head, elapsed time, and mass balance parameters from cards
	41-44	A4	IPU2	PUN2 to punch computed head, elapsed time, and mass balance parameters on cards
	46-49	A4	ITK	ITKR to read the value of TK(I,J,K) for simulations in which confining layers are not represented by layers of nodes. $TK(I,J,K) = K_{zz}/b$.

Group II: Scalar parameters

The parameters required in every problem are underlined. The other parameters are required as noted; when not required, their location on the card can be left blank. The G format is used to read E, F and I format data. Minimize mistakes by always right-justifying data in the field. If F format data do not contain significant figures to the right of the decimal point, the decimal point can be omitted.

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
1	1-10	G10.0	NPER	Number of pumping periods for the simulation
	11-20	G10.0	KTH	Number of time steps between printouts
	Note: To print only the results for the final time step in a pumping period, make KTH greater than the expected number of time steps. The program always prints the results for the final time step.			
	21-30	G10.0	ERR	Error criteria for closure (L)
	Note: When the head change at all nodes on subsequent iterations is less than this value (for example, 0.01 foot), the program has converged to a solution for the time step.			
	31-40	G10.0	LENGTH	Number of iteration parameters
2	1-10	G10.0	XSCALE	Factor to convert model length unit to unit used in X direction on maps (e.g. to convert from feet to miles, XSCALE = 5280). For no maps, card 2 is blank
	11-20	G10.0	YSCALE	Factor to convert model length unit to unit used in Y direction on maps.
	21-30	G10.0	DINCH	Number of map units per inch
	31-40	G10.0	FACT1	Factor to adjust value of drawdown printed*
	41-49	9I1	LEVEL1(I)	Layers for which drawdown maps are to be printed. List the layers startir in columun 41; the first zero entry terminates the printing of drawdown maps.

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
2	51-60	G10.0	FACT2	Factor to adjust value of head printed*
	61-69	9I1	LEVEL2(I)	Layers for which head maps are to be printed. List layers starting in column 61; the first zero entry terminates the printing of head maps.
	71-78	A8	MESUR	Name of map length unit.
3	1-20	G20.10	SUM	Parameters in which elapsed time and cumulative volumes for mass balance are stored. For the start of a simulation insert three blank cards. For continuation of a previous run using cards as input, replace the three blank cards with the first three cards of punched output from the previous run. Using data from disk for input, leave the three blank cards in the data deck.
	21-40	G20.10	SUMP	
	41-60	G20.10	PUMPT	
	61-80	G20.10	CFLUXT	
4	1-20	G20.10	QRET	
	21-40	G20.10	CHST	
	41-60	G20.10	CHDT	
	61-80	G20.10	FLUXT	
5	1-20	G20.10	STORT	
	21-40	G20.10	ETFLXT	
	41-60	G20.10	FLXNT	

*Value of drawdown or head	FACT 1 or FACT 2	Printed value
52.57	0.01	1
	0.1	5
	1.0	53
	10.0	526
	100.0	***

Group III: Array Data

Each of the following data sets (except data set 1) consists of a parameter card and, if the data set contains variable data, a set of data cards. If the data set requires data for each layer, a parameter card and data cards (for layers with variable data) are required for each layer. Each parameter card contains at least five variables:

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
Every Parameter Card	1-10	G10.0	FAC	If IVAR = 0, FAC is the value assigned to every element of the matrix for this layer. If IVAR = 1, FAC is the multiplication factor for the following set of data cards for this layer.
	11-20	G10.0	IVAR	= 0 if no data cards are to be read for this layer. = 1 if data cards for this layer follow.
	21-30	G10.0	IPRN	= 0 if input data for this layer are to be printed; = 1 if input data for the layer are <u>not</u> to be printed.
Trans-missivity Parameter Cards also have these Variables	31-40	G10.0	FACT(K,1)	multiplication factor for transmissivity in x direction
	41-50	G10.0	FACT(K,2)	multiplication factor for transmissivity in the y direction
	51-60	G10.0	FACT(K,3)	multiplication factor for hydraulic conductivity in the z direction. (Not used when confining bed nodes are eliminated and TK values are read)
Every Parameter Card	61-70	G10.0	IRECS	= 0 if the matrix is being read from cards or if each element is being set equal to FAC . = 1 if the matrix is to be read from disk (unit 2)
	71-80	G10.0	IRECD	= 0 if the matrix is <u>not</u> to be stored on disk. = 1 if the matrix being read from cards or set equal to FAC <u>is</u> to be stored on disk (unit 2) for later retrieval.

When data cards are included, start each row on a new card. To prepare a set of data cards for an array that is a function of space, the general procedure is to overlay the finite-difference grid on a contoured map of the parameter and record the average value of the parameter for each finite-difference block on coding forms according to the appropriate format. In general, record only significant digits and no decimal points (except for data set 2); use the multiplication factor to convert the data to their appropriate values. For example, if DELX ranges from 1000 to 15000 feet, coded values should range from 1-15; the multiplication factor (FAC) would be 1000.

DATA SET	COLUMNS	FORMAT	VARIABLE	DEFINITION
1	1-80	8F 10.4	PHI(I,J,K)	Head values for continuation of a previous run (L)

Note: For a new simulation this data set is omitted. Do not include a parameter card with this data set.

2	1-80	8F 10.4	STRT(I,J,K)	Starting head matrix (L)
3	1-80	20F 4.0	S (I,J,K)	Storage coefficient (dimensionless)

Note: This matrix is also used to locate constant head boundaries by coding a negative number at constant head nodes. At these nodes T must be greater than zero. If equation 3 is to be solved, read specific storage instead of storage coefficient.

4	1-80	20F 4.0	T(I,J,K)	Transmissivity (L^2/t)

Note 1) Zero values are required around the perimeter of the T matrix for each layer for reasons inherent in the computational scheme. This is done automatically by the program.

2) See the previous page for the additional requirements on the parameter cards for this data set.

3) If the upper active layer is unconfined and PERM and BOTTOM are to be read for this layer, insert a parameter card <u>for this layer</u> with only the values for FACT on it. If equation 3 is to be solved read hydraulic conductivity insteady of transmissivity.

5	1-80	20F 4.0	TK(I,J,K)	K_{zz}/b

Note: This data set is read only if specified in the options. The number of layers of TK values = K'- 1. See the discussion of the treatment of confining layers.

6	1-80	20F 4.0	PERM(I,J)	Hydraulic conductivity (L/T) (see note 1 for data set 4)
7	1-80	20F 4.0	BOTTOM(I,J)	Elevation of bottom of water-table unit (L)

DATA SET	COLUMNS	FORMAT	VARIABLE	DEFINITION

Note: Data sets 6 and 7 are required only for simulating unconfined conditions in the upper hydrologic unit.

DATA SET	COLUMNS	FORMAT	VARIABLE	DEFINITION
8	1-80	20F 4.0	QRE(I,J)	Recharge rate (L/T)

Note: Omit if not used

DATA SET	COLUMNS	FORMAT	VARIABLE	DEFINITION
9	1-80	8G10.0	DELX(J)	Grid spacing in x direction (L)
10	1-80	8G10.0	DELY(I)	Grid spacing in y direction (L)
11	1-80	8G10.0	DELZ(K)	Grid spacing in z direction (L)

Group IV: Parameters that change with the pumping period

The program has two options for the simulation period:

1. To simulate a given number of time steps, set TMAX to a value larger than the expected simulation period. The program will use NUMT, CDLT, and DELT as coded. If NUMT is greater than 50 change the dimension of ITT0 in subroutine STEP to the appropriate size.
2. To simulate a given pumping period, set NUMT larger than the number required for the simulation period (for example, 50). The program will compute the exact DELT (which will be ≤ DELT coded) and NUMT to arrive exactly at TMAX on the last time step.

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
1	1-10	G10.0	KP	Number of the pumping period
	11-20	G10.0	KPM1	Number of the previous pumping period

Note: KPM1 is currently not used

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
	21-30	G10.0	NWEL	Number of wells for this pumping period
	31-40	G10.0	TMAX	Number of days in this pumping period
	41-50	G10.0	NUMT	Number of time steps
	51-60	G10.0	CDLT	Multiplying factor for DELT

Note: 1.5 is commonly used

CARD	COLUMNS	FORMAT	VARIABLE	DEFINITION
	61-70	G10.0	DELT	Initial time step in hours

If NWEL: 0 the following set of cards is omitted

DATA SET 1 (NWEL cards)

COLUMNS	FORMAT	VARIABLE	DEFINITION
1-10	G10.0	K	Layer in which well is located
11-20	G10.0	I	Row location of well
21-30	G10.0	J	Column location of well
31-40	G10.0	WELL(I,J,K)	Pumping rate (L^3/t), negative for a pumping well.

For each additional pumping period, another set of group IV cards is required (that is, NPER sets of group IV cards are required).

APPENDIX IV

EXAMPLE SIMULATION

The following pages illustrate the data input and results of the steady-state simulation of a hypothetical problem including two aquifers separated by a confining layer (figure 11). Boundaries are no-flow except for a constant-head boundary along the left side of the upper aquifer. The aquifers are identical except that the upper aquifer is unconfined. The confining bed is not represented by a layer of nodes because only vertical flow is simulated and this is incorporated in the equations for the two aquifers. The finite-difference grid, therefore, consists of two layers of nodes with uniform grid spacing. There are two discharging wells, one in each of the two aquifers.

Table 4 lists the data input required for this simulation. The format for data are given in Appendix III. The location of numbers on the data cards should not be difficult to determine because they are either in fields of 4 or 10 spaces. In general, zero values have not been coded.

The printout of this simulation follows Table 4 and is either self-explanatory or has been discussed in Appendix II.

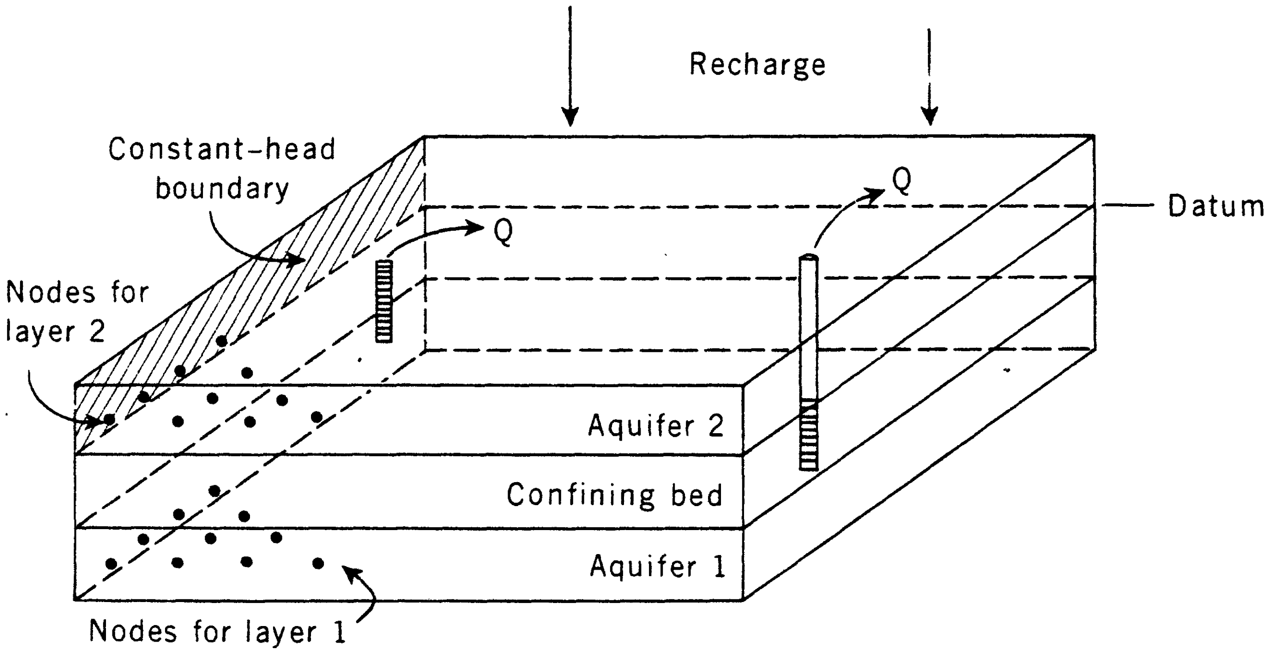

Figure 11.--Schematic illustration of example problem.

TABLE 4.--DATA INPUT FOR EXAMPLE PROBLEM.

GROUP I					---------PUMPING FROM 2 AQUIFERS, UPPER AQU				
		IFER HAS A FREE SURFACE---------							
		20	20	2	50	18			
		DRAW MASS		WATE RECH		ITKR			
GROUP II		1	1	.001	5				
		1000	1000	2	1012				1000FEFT
GROUP III	STRT	100							
		100							
	S	1	1						
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
		-1							
	T	.1			1	1			
		.1			1	1			
	TK	.00000001							
	PERM	.001							
	BOTTOM								
	QRE	1.0F-09							
	DELX	1000							
	DELY	1000							
	DELZ	1							
GROUP IV		1	0	2	1	1	1	24	
		1	10	5	-1				
		2	10	15	-1				

```
                      ---------PUMPING FROM 2 AQUIFERS, UPPER AQUIFER HAS A FREE SURFACE---------

                                                     NUMBER OF ROWS =   20
                                                  NUMBER OF COLUMNS =   20
                                                   NUMBER OF LAYERS =    2

                             MAXIMUM PERMITTED NUMBER OF ITERATIONS =   50

                                      NUMBER OF CONSTANT HEAD NODES =   18

SIMULATION OPTIONS: DRAW          MASS                WATE    RECH                 ITKR

                                            WORDS OF VECTOR Y USED =  12992

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

                                        ERROR CRITERIA FOR CLOSURE =  0.9999999E-03


                    ON ALPHAMERIC MAP:
                            MULTIPLICATION FACTOR FOR X DIMENSION =   1000.000
                            MULTIPLICATION FACTOR FOR Y DIMENSION =   1000.000
                                            MAP SCALE IN UNITS OF    1000FEET
                                      NUMBER OF 1000FEET PER INCH =   2.000000
                              MULTIPLICATION FACTOR FOR DRAWDOWN =   10.00000     PRINTED FOR LAYERS 1 2 0 0 0 0 0 0 0
                                  MULTIPLICATION FACTOR FOR HEAD =  0.0          PRINTED FOR LAYERS 0 0 0 0 0 0 0 0 0

                                                   STARTING HEAD =   100.0000     FOR LAYER  1

                                                   STARTING HEAD =   100.0000     FOR LAYER  2

                                             STORAGE COEFFICIENT =  0.0          FOR LAYER  1
```

```
                                         STORAGE COEFFICIENT MATRIX, LAYER  2
                                   ----------------------------------------

  1  0.0      0.0        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  2  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  3  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  4  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  5  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  6  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  7  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  8  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

  9  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 10  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 11  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 12  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 13  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 14  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 15  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 16  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 17  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 18  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 19  0.0     -1.00000    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0      0.0        0.0     0.0     0.0     0.0

 20  0.0      0.0        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
```

```
 0.0       0.0       0.0       0.0       0.0       0.0
                                                  TRANSMISSIVITY =  0.9999996E-01 FOR LAYER  1
                                    DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORS FOR LAYER  1
                                                                 X =   1.000000
                                                                 Y =   1.000000
                                                                 Z =  0.0
                                                  TRANSMISSIVITY =  0.9999996E-01 FOR LAYER  2
                                    DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORS FOR LAYER  2
                                                                 X =   1.000000
                                                                 Y =   1.000000
                                                                 Z =  0.0
                                                              TK =  0.9999997E-08 FOR LAYER  1
                                          HYDRAULIC CONDUCTIVITY =  0.9999999E-03 FOR LAYER  2
                                                BOTTOM ELEVATION =  0.0           FOR LAYER  2
                                                   RECHARGE RATE =  0.9999999E-09 FOR LAYER  2
                                                            DELX =   1000.000
                                                            DELY =   1000.000
                                                            DELZ =   1.000000

                                    SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE
                                    -------------------------------------------

5 ITERATION PARAMETERS:  0.0        0.7045910D 00  0.9127335D 00  0.9742207D 00  0.9923846D 00

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

                                        NUMBER OF TIME STEPS=     1

                                           DELT IN HOURS =    24.000

                                        MULTIPLIER FOR DELT =     1.000

                                                  2 WELLS
                                                  -------

                                       K        I         J    PUMPING RATE

                                       1       10         5        -1.00
                                       2       10        15        -1.00
```

```
                                          ---------------------------------------------------------
                                          |              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 PERIOC IN DAYS=          1.00
                                                                             YEARS=          0.00


          CUMULATIVE MASS BALANCE:                L**3                     RATES FOR THIS TIME STEP:               L**3/T
          ------------------------                                         -------------------------

                   SOURCES:                                                                   STORAGE =             0.0
                   --------                                                                  RECHARGE =             0.3060
                          STORAGE =              0.0                                    CONSTANT FLUX =             0.0
                         RECHARGE =         26438.11                                          PUMPING =            -2.0000
                    CONSTANT FLUX =              0.0                           EVAPOTRANSPIRATION =             0.0
                    CONSTANT HEAD =        145961.88                                   CONSTANT HEAD:
                          LEAKAGE =              0.0                                               IN =             1.6894
                    TOTAL SOURCES =        172399.94                                              OUT =             0.0
                                                                                            LEAKAGE:
                   DISCHARGES:                                          FROM PREVIOUS PUMPING PERIOD =             0.0
                   -----------                                                                  TOTAL =             0.0
               EVAPOTRANSPIRATION =              0.0
                    CONSTANT HEAD =              0.0                                     SUM OF RATES =            -0.0046
                  QUANTITY PUMPED =        172799.81
                          LEAKAGE =              0.0
                  TOTAL DISCHARGE =        172799.81

                DISCHARGE-SOURCES =           399.88
               PER CENT DIFFERENCE =             0.23


     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)
          -    -    -      -------------         -    -    -      -------------         -    -    -      -------------


          2    2    2      0.8160883E-01         2    3    2      0.8287251E-01         2    4    2      0.8543676E-01
          2    5    2      0.8936179E-01         2    6    2      0.9468210E-01         2    7    2      0.1012863
          2    8    2      0.1086478             2    9    2      0.1153280             2   10    2      0.1184522
          2   11    2      0.1147698             2   12    2      0.1075212             2   13    2      0.9957021E-01
          2   14    2      0.9234345E-01         2   15    2      0.8635312E-01         2   16    2      0.8169365E-01
          2   17    2      0.7830906E-01         2   18    2      0.7611233E-01         2   19    2      0.7503170E-01

     FLOW TO TOP LAYER = -0.9975219      FLOW TO BOTTOM LAYER = -0.9975219        POSITIVE UPWARD

     MAXIMUM HEAD CHANGE FOR EACH ITERATION:
     ---------------------------------------
```

```
     3.4372     0.7898     0.5177     0.6253     1.3647     0.0958     0.0958     0.2184     0.3409     0.7132
     0.0415     0.0415     0.0936     0.1805     0.4106     0.0268     0.0266     0.0605     0.1020     0.2055
     0.0122     0.0122     0.0275     0.0532     0.1209     0.0079     0.0078     0.0178     0.0301     0.0605
     0.0036     0.0036     0.0081     0.0157     0.0356     0.0023     0.0023     0.0052     0.0088     0.0178
     0.0011     0.0011     0.0024     0.0046     0.0104     0.0007
TIME STEP :  1
----------
ITERATIONS: 45
----------
```

```
                         PLOT OF DRAWDOWN          LAYER   1

+---------+---------+---------+---------+---------+---------+---------+---------+---------+   18.00
I 45  45  46  47  47  48  49  50  50  49  48  47  46  45  44  44  43  43 I
I 45  45  46  46  47  48  49  50  50  49  49  47  46  45  44  43  43  43 I
+                                                                        +   16.00
I 44  45  45  46  47  48  49  50  50  50  48  47  46  45  44  43  42  42 I
I 44  44  45  45  46  47  49  50  50  50  48  47  45  44  43  42  42  41 I
+                                                                        +   14.00
I 43  43  44  44  45  47  48  49  50  49  47  46  44  43  42  41  41  40 I
I 41  42  42  43  44  45  46  48  48  47  46  44  43  42  41  40  39  39 I
+                                                                        +   12.00
I 40  40  41  41  42  43  44  45  46  45  44  43  41  40  39  38  38  38 I
I 38  38  39  40  41  41  42  43  43  43  42  41  40  38  38  37  36  36 I
+                                                                        +   10.00
I 36  37  37  38  39  39  40  41  41  40  40  39  38  37  36  35  34  34 I  X DIS-
I                                                                        I TANCE IN
I 34  34  35  36  37  37  38  39  39  39  38  37  36  35  34  33  32  32 I 1000FEET
+                                                                        +    8.00
I 32  32  33  34  35  36  37  38  38  37  36  35  34  33  31  31  30  30 I
I 30  30  30  31  33  34  36  37  38  37  35  34  32  30  29  28  28  28 I
+                                                                        +    6.00
I 27  27  28  29  31  33  35  38  40  38  35  32  30  28  27  26  25  25 I
I 25  25  26  27  29  31  35  41  47  41  35  31  28  26  25  24  23  23 I
+                                                                        +    4.00
I 22  22  23  24  26  30  35  45  68  44  34  29  26  24  22  21  21  20 I
I 20  20  21  22  24  27  31  36  42  36  30  26  23  21  20  19  18  18 I
+                                                                        +    2.00
I 18  18  19  20  21  24  27  30  32  29  26  23  21  19  18  17  16  16 I
I 16  17  17  18  20  22  24  26  27  26  24  21  19  18  17  16  15  15 I
+---------+---------+---------+---------+---------+---------+---------+---------+---------+   0.0

0.0      2.00      4.00      6.00      8.00     10.00     12.00     14.00     16.00     18.00
```

```
                         DISTANCE FROM ORIGIN IN Y DIRECTION, IN 1000FEET
EXPLANATION
-----------

R = CONSTANT HEAD BOUNDARY
*** = VALUE EXCEEDED 3 FIGURES
MULTIPLICATION FACTOR =  10,000
```

```
                         PLOT OF DRAWDOWN       LAYER   2

+---------+---------+---------+---------+---------+---------+---------+---------+---------+  18.00
I                                                                                        I
I 45  46  46  47  48  50  51  53  53  52  51  49  47  46  45  44  41  41 I
I                                                                                        I
I 45  45  46  47  48  50  52  54  54  53  52  49  47  46  44  43  43  42 I
+                                                                                        +  16.00
I 44  45  46  47  48  51  53  56  58  56  53  50  47  45  44  43  42  42 I
I                                                                                        I
I 44  44  45  46  48  51  55  61  67  61  55  50  47  45  43  42  41  41 I
+                                                                                        +  14.00
I 43  43  44  45  47  50  56  66  92  66  55  50  46  44  42  41  40  40 I
I                                                                                        I
I 41  42  42  44  46  48  52  58  65  58  52  48  45  42  41  40  39  39 I
+                                                                                        +  12.00
I 40  40  41  42  43  45  48  51  53  51  48  45  42  41  39  38  38  37 I
I                                                                                        I
I 38  38  39  40  41  42  44  46  46  45  44  42  40  38  37  36  36  35 I
+                                                                                        +  10.00
I                                                                                        I   X DIS-
I 36  36  36  37  38  39  40  41  42  41  40  39  37  36  35  34  34  33 I  TANCE IN
I                                                                                        I
I 33  33  34  35  35  36  37  38  38  38  37  36  35  34  33  32  31  31 I  1000FEET
+                                                                                        +   8.00
I 31  31  31  32  33  34  34  35  35  35  34  33  32  31  30  29  29  29 I
I                                                                                        I
I 28  28  28  29  30  31  31  32  32  32  31  30  29  28  27  27  26  26 I
+                                                                                        +   6.00
I 24  25  25  26  27  28  29  29  30  29  28  27  26  25  24  23  23  23 I
I                                                                                        I
I 21  21  21  22  23  24  25  26  27  26  25  24  22  21  21  20  20  19 I
+                                                                                        +   4.00
I 17  17  17  18  19  20  21  22  23  22  21  19  18  17  17  16  16  16 I
I                                                                                        I
I 12  12  13  13  14  14  15  16  17  16  15  14  13  13  12  12  11  11 I
+                                                                                        +   2.00
I  7   7   7   7   8   8   9   9   9   9   8   8   7   7   7   6   6   6 I
I                                                                                        I
I  R   R   R   R   R   R   R   R   R   R   R   R   R   R   R   R   R   R I
+---------+---------+---------+---------+---------+---------+---------+---------+---------+   0.0

0.0      2.00      4.00      6.00      8.00     10.00     12.00     14.00     16.00     18.00
```

```
                          DISTANCE FROM ORIGIN IN Y DIRECTION, IN 1000FEET
EXPLANATION
-----------

R = CONSTANT HEAD BOUNDARY
*** = VALUE EXCEEDED 3 FIGURES
MULTIPLICATION FACTOR =  10.000
```

```
                                          DRAWDOWN, LAYER  1
                                          ------------------

  1  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0     0.0

  2  0.0     1.63    1.76    1.96    2.20    2.45    2.71    2.96    3.20    3.42    3.63    3.82    4.00    4.15    4.28    4.38    4.45    4.50
     4.52    0.0

  3  0.0     1.66    1.79    2.00    2.24    2.49    2.74    2.99    3.22    3.45    3.66    3.85    4.02    4.18    4.30    4.40    4.48    4.52
     4.55    0.0

  4  0.0     1.73    1.86    2.07    2.32    2.57    2.81    3.05    3.28    3.50    3.70    3.90    4.07    4.23    4.36    4.46    4.53    4.57
     4.59    0.0

  5  0.0     1.84    1.98    2.20    2.45    2.69    2.92    3.14    3.36    3.57    3.77    3.97    4.15    4.31    4.44    4.53    4.60    4.64
     4.66    0.0

  6  0.0     1.99    2.15    2.39    2.65    2.87    3.07    3.26    3.45    3.65    3.85    4.05    4.24    4.40    4.54    4.63    4.69    4.73
     4.74    0.0

  7  0.0     2.18    2.37    2.66    2.95    3.14    3.28    3.41    3.57    3.74    3.94    4.14    4.34    4.52    4.66    4.75    4.80    4.82
     4.83    0.0

  8  0.0     2.39    2.65    3.06    3.47    3.53    3.53    3.58    3.68    3.83    4.02    4.23    4.44    4.64    4.80    4.87    4.90    4.90
     4.90    0.0

  9  0.0     2.59    2.95    3.63    4.45    4.09    3.81    3.72    3.76    3.89    4.07    4.29    4.52    4.75    4.93    4.98    4.98    4.96
     4.95    0.0

 10  0.0     2.68    3.15    4.25    6.84    4.70    4.00    3.79    3.79    3.91    4.09    4.31    4.55    4.80    5.02    5.03    5.01    4.98
     4.96    0.0

 11  0.0     2.58    2.94    3.62    4.44    4.07    3.79    3.70    3.74    3.87    4.05    4.26    4.50    4.73    4.90    4.96    4.95    4.94
     4.93    0.0

 12  0.0     2.37    2.63    3.04    3.44    3.50    3.50    3.54    3.64    3.79    3.97    4.18    4.39    4.59    4.74    4.82    4.85    4.85
     4.85    0.0

 13  0.0     2.14    2.33    2.62    2.91    3.09    3.22    3.35    3.50    3.68    3.87    4.07    4.27    4.44    4.58    4.67    4.72    4.74
     4.75    0.0

 14  0.0     1.94    2.10    2.33    2.59    2.81    3.00    3.18    3.37    3.56    3.76    3.96    4.14    4.30    4.44    4.53    4.59    4.62
     4.63    0.0

 15  0.0     1.78    1.92    2.13    2.37    2.60    2.83    3.04    3.25    3.45    3.66    3.85    4.02    4.18    4.31    4.40    4.47    4.51
     4.53    0.0

 16  0.0     1.65    1.78    1.98    2.22    2.46    2.70    2.93    3.15    3.36    3.56    3.75    3.92    4.08    4.20    4.30    4.37    4.41
     4.43    0.0

 17  0.0     1.56    1.69    1.88    2.12    2.36    2.61    2.84    3.07    3.29    3.49    3.68    3.85    4.00    4.12    4.22    4.29    4.34
     4.36    0.0

 18  0.0     1.51    1.63    1.82    2.05    2.30    2.55    2.79    3.02    3.24    3.44    3.63    3.80    3.94    4.07    4.16    4.24    4.28
     4.31    0.0

 19  0.0     1.48    1.60    1.79    2.02    2.27    2.52    2.76    2.99    3.21    3.42    3.60    3.77    3.92    4.04    4.14    4.21    4.26
     4.28    0.0

 20  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0     0.0
```

```
                                                  DRAWDOWN, LAYER  2
                                                  -----------------

   1  0.0    0.0    0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
      0.0    0.0

   2  0.0    0.0    0.66  1.20  1.67  2.08  2.44  2.77  3.06  3.32  3.56  3.78  3.97  4.13  4.27  4.37  4.45  4.49
      4.52   0.0

   3  0.0    0.0    0.66  1.22  1.69  2.11  2.47  2.79  3.08  3.34  3.59  3.81  4.00  4.17  4.31  4.41  4.48  4.53
      4.55   0.0

   4  0.0    0.0    0.68  1.25  1.73  2.15  2.51  2.83  3.13  3.39  3.64  3.87  4.07  4.25  4.39  4.49  4.56  4.60
      4.62   0.0

   5  0.0    0.0    0.71  1.30  1.80  2.21  2.58  2.90  3.19  3.46  3.72  3.96  4.18  4.38  4.52  4.62  4.68  4.71
      4.72   0.0

   6  0.0    0.0    0.75  1.37  1.88  2.30  2.66  2.98  3.27  3.55  3.82  4.09  4.34  4.56  4.73  4.82  4.85  4.85
      4.85   0.0

   7  0.0    0.0    0.80  1.45  1.98  2.40  2.75  3.06  3.35  3.64  3.93  4.24  4.55  4.84  5.05  5.10  5.07  5.02
      4.99   0.0

   8  0.0    0.0    0.85  1.54  2.09  2.51  2.85  3.15  3.43  3.73  4.05  4.40  4.81  5.24  5.58  5.51  5.34  5.21
      5.14   0.0

   9  0.0    0.0    0.90  1.63  2.22  2.62  2.93  3.21  3.49  3.79  4.13  4.56  5.10  5.82  6.61  6.09  5.64  5.37
      5.25   0.0

  10  0.0    0.0    0.92  1.68  2.31  2.67  2.96  3.22  3.50  3.80  4.16  4.62  5.29  6.47  9.20  6.74  5.83  5.45
      5.29   0.0

  11  0.0    0.0    0.89  1.62  2.20  2.60  2.91  3.19  3.47  3.77  4.11  4.53  5.08  5.80  6.59  6.06  5.61  5.35
      5.23   0.0

  12  0.0    0.0    0.84  1.52  2.07  2.49  2.82  3.11  3.39  3.68  4.00  4.36  4.76  5.19  5.52  5.45  5.29  5.15
      5.09   0.0

  13  0.0    0.0    0.78  1.42  1.94  2.36  2.71  3.01  3.29  3.58  3.86  4.17  4.47  4.76  4.97  5.02  4.99  4.94
      4.91   0.0

  14  0.0    0.0    0.73  1.33  1.83  2.24  2.59  2.90  3.19  3.46  3.73  3.99  4.24  4.46  4.63  4.71  4.74  4.74
      4.74   0.0

  15  0.0    0.0    0.69  1.26  1.74  2.14  2.49  2.80  3.09  3.35  3.60  3.84  4.06  4.25  4.39  4.49  4.54  4.57
      4.58   0.0

  16  0.0    0.0    0.65  1.20  1.66  2.06  2.41  2.72  3.00  3.26  3.50  3.72  3.92  4.09  4.23  4.33  4.40  4.44
      4.45   0.0

  17  0.0    0.0    0.63  1.16  1.61  2.00  2.34  2.65  2.93  3.19  3.42  3.63  3.82  3.98  4.12  4.22  4.29  4.33
      4.36   0.0

  18  0.0    0.0    0.61  1.13  1.57  1.96  2.30  2.61  2.89  3.14  3.37  3.58  3.76  3.92  4.05  4.15  4.22  4.27
      4.29   0.0

  19  0.0    0.0    0.60  1.11  1.55  1.94  2.28  2.59  2.86  3.12  3.34  3.55  3.73  3.88  4.01  4.11  4.19  4.24
      4.26   0.0

  20  0.0    0.0    0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
      0.0    0.0
```

APPENDIX V

DEFINITION OF PROGRAM VARIABLES

A	DUMMY ARRAY NAME USED TO INPUT MATRIX DATA;
B	TC(I-1,J,K)/DELY(I);
BOTTOM	ELEVATION OF THE BOTTOM OF THE UPPER UNIT;
COLT	MULTIPLYING FACTOR FOR THE TIME STEP;
D	TR(I,J-1,K)/DELX(J);
DDN	VECTOR THAT CONTAINS DRAWDOWN VALUES (L);
DELT	TIME INCREMENT (T);
DELX	GRID SPACING IN THE X DIRECTION (L);
DELY	GRID SPACING IN THE Y DIRECTION (L);
DELZ	GRID SPACING IN THE Z DIRECTION (L);
DLM	DUMMY ARRAY USED TO COMPLETE THE ARGUMENT LIST FOR ENTRY ARRAY WHEN THE LAST 3 ELEMENTS ARE NOT REQUIRED;
FL	ELEMENT OF UPPER TRIANGULAR FACTOR U;
ERR	CLOSURE CRITERIA (L);
F	TR(I,J,K)/DELX(J);
FAC	SEE EXPLANATION IN GROUP III: ARRAY DATA;
FL	ELEMENT OF UPPER TRIANGULAR FACTOR U;
FLOW	FLOW RATE TO A CONSTANT-HEAD NODE (L**3/T);
GL	ELEMENT OF UPPER TRIANGULAR FACTOR U;
H	TC(I,J,K)/DELY(I);
HEADNG	TITLE FOR SIMULATION;
I0	NUMBER OF ROWS;
ICHK	VECTOR CONTAINING PROBLEM OPTIONS;
IDK1	OPTION TO READ HEAD DATA FROM DISK;
IDK2	OPTION TO WRITE RESULTS ON DISK;
IDRAW	OPTION TO PRINT DRAWDOWN;
IERR	= 2, PROGRAM HAS EXCEEDED PERMITTED ITERATIONS;
IFINAL	= 0 ALL TIME STEPS EXCEPT THE LAST; = 1 LAST TIME STEP IN PUMPING PERIOD;
IFLO	OPTION TO COMPUTE A VOLUMETRIC BALANCE;
IHEAD	OPTION TO PRINT HEAD MATRIX;
IK,JK,K5	DIMENSIONS OF TK ARRAY;
IMAX	MAXIMUM OF I0,J0;
IN	DUMMY ARRAY USED TO PRINT MATRIX NAMES IN DATAI;
INFT	AN ARRAY OF FORMATS USED TO READ MATRIX DATA;
IOFT	AN ARRAY OF FORMATS USED TO PRINT MATRIX DATA;
IP,JP	DIMENSIONS OF PERM AND BOTTOM ARRAYS;
IPRN	SEE EXPLANATION IN GROUP III: ARRAY DATA;
IPU1	OPTION TO READ HEAD AND MASS BALANCE VALUES FROM CARDS;
IPU2	OPTION TO PUNCH RESULTS ON CARDS;
IQ,JQ	DIMENSIONS OF QRE ARRAY;
IQRE	OPTION FOR RECHARGE;
ISUM	THE CUMULATIVE WORDS OF STORAGE USED IN THE VECTOR Y;
IRECD	SEE EXPLANATION IN GROUP III: ARRAY DATA;
IRECS	SEE EXPLANATION IN GROUP III: ARRAY DATA;
IRN	COUNTER TO INDICATE THE CURRENT RECORD NUMBER IN THE DIRECT ACCESS DATA SET;
IT	ITERATION COUNTER;
ITMAX	MAXIMUM NUMBER OF ITERATIONS PER TIME STEP;
ITMX1	ITMAX+1;
ITTO	VECTOR CONTAINING TOTAL NUMBER OF ITERATIONS PER TIME STEP;

IVAR	SEE EXPLANATION IN GROUP III: ARRAY DATA;
IWATER	OPTION FOR WATER-TABLE CONDITIONS IN UPPER LAYER;
J0	NUMBER OF COLUMNS;
JFLO	ARRAY CONTAINING LOCATION OF CONSTANT-HEAD NODES;
K0	NUMBER OF LAYERS;
KP	NUMBER OF THE PUMPING PERIOD;
KPM1	NOT CURRENTLY USED;
KT	TIME STEP COUNTER;
KTH	NUMBER OF TIME STEPS BETWEEN PRINTOUTS;
L	VECTOR CONTAINING INITIAL ADDRESS OF ARRAYS;
LENGTH	NUMBER OF ITERATION PARAMETERS;
LEVEL1	VECTOR CONTAINING LAYERS FOR WHICH DRAWDOWN MAPS ARE TO BE PRINTED;
LEVEL2	VECTOR CONTAINING LAYERS FOR WHICH HEAD MAPS ARE TO BE PRINTED;
NAME	AN ARRAY OF MATRIX NAMES;
NCH	NUMBER OF CONSTANT-HEAD NODES;
NPER	NUMBER OF PUMPING PERIODS;
NUMT	NUMBER OF TIME STEPS;
NWEL	NUMBER OF WELLS FOR A PUMPING PERIOD;
OLD	HEAD AT THE END OF THE PREVIOUS TIME STEP;
PERM	HYDRAULIC CONDUCTIVITY OF THE UPPER UNIT;
PHI	HYDRAULIC HEAD (L);
QR	RECHARGE RATE(L/T);
QRE	RECHARGE RATE (L/T);
RHO	S/DELT (1/T);
RHOP	VECTOR CONTAINING ITERATION PARAMETERS;
S	STORAGE COEFFICIENT;
STRT	HYDRAULIC HEAD AT THE START OF THE SIMULATION;
SU	TK(I,J,K)/DELZ(K);
SUM	TOTAL ELAPSED TIME IN THE SIMULATION (T);
SUMP	TOTAL ELAPSED TIME IN THE PUMPING PERIOD (T);
T	TRANSMISSIVITY (L**2/T);
TC	HARMONIC AVERAGE OF T/DELY @ I+1/2,J,K (L/T);
TEST	= 0 CLOSURE CRITERIA SATISFIED; = 1 CLOSURE CRITERIA NOT SATISFIED;
TEST3	MAXIMUM CHANGE IN HEAD FOR THE TIME STEP;
TF	AN ARRAY USED TO READ AND TRANSFER DIRECTIONAL TRANSMISSIVITY FACTORS;
TK	HARMONIC AVERAGE OF BK/DELZ @ I,J,K+1/2 (L/T);
TMAX	NUMBER OF DAYS IN THE PUMPING PERIOD (T);
TR	HARMONIC AVERAGE OF T/DELX @ I,J+1/2,K (L/T);
V	INTERMEDIATE VECTOR;
WELL	WELL DISCHARGE (L**3/T);
XI	ARRAY CONTAINING INCREMENTAL HEAD VALUES IN SIP SOLUTION (L);
Y	VECTOR CONTAINING ARRAY STORAGE;
Z	TK(I,J,K-1)/DELZ(K).

DEFINITION OF VARIABLES IN CHECKI SUBROUTINE

CFLUX	INFLOW FROM RECHARGE WELLS (L**3/T);
CFLUXT	CUMULATIVE VOLUME OF WATER FROM RECHARGE WELLS (L**3);
CHD1	RATE OF OUTFLOW TO CONSTANT HEAD BOUNDARY (L**3/T);
CHD2	RATE OF INFLOW FROM CONSTANT HEAD BOUNDARY (L**3/T);
CHDT	CUMULATIVE DISCHARGE TO CONSTANT HEAD BOUNDARY (L**3);

CHST	CUMULATIVE VOLUME OF WATER INFLOW FROM CONSTANT HEAD BOUNDARY (L**3);
DIFF	ERROR IN MASS BALANCE (L**3);
ETFLUX	EVAPOTRANSPIRATION RATE (L**3/T);
ETFLXT	CUMULATIVE DISCHARGE BY ET (L**3);
FLUX	RATE OF LEAKAGE DUE TO GRADIENTS AT THE START OF THE PUMPING PERIOD (L**3/T);
FLUXS	NET LEAKAGE RATE (L**3/T);
FLXN	RATE OF DISCHARGE BY LEAKAGE (L**3/T);
FLXNT	CUMULATIVE VOLUME OF WATER DISCHARGED BY LEAKAGE (L**3);
FLXPT	CUMULATIVE VOLUME OF WATER INFLOW FROM LEAKAGE (L**3);
PERCNT	PERCENT ERROR IN CUMULATIVE MASS BALANCE;
PUMP	DISCHARGE FROM WELLS (L**3/T);
PUMPT	CUMULATIVE VOLUME OF WATER DISCHARGED BY PUMPING WELLS (L**3);
QREFLX	RECHARGE RATE (L**3/T);
QRET	CUMULATIVE VOLUME OF WATER DERIVED FROM RECHARGE (L**3);
STOR	RATE OF CHANGE IN STORAGE FOR THE TIME STEP (L**3/T);
STORT	CUMULATIVE VOLUME OF WATER DERIVED FROM STORAGE (L**3);
SUMR	SUM OF RECHARGE AND DISCHARGE RATES FOR THE TIME STEP (L**3/T);
TOTL1	CUMULATIVE VOLUME OF WATER FROM ALL SOURCES (L**3);
TOTL2	CUMULATIVE VOLUME OF WATER DISCHARGED FROM THE SYSTEM (L**3);
X	NET FLOW TO BOTTOM LAYER (L**3/T);
Y	NET FLOW TO TOP LAYER (L**3/T).

DEFINITION OF VARIABLES IN THE PRINTAI SUBROUTINE

BLANK	CONTAINS BLANK SYMBOLS;
DINCH	NUMBER OF MAP UNITS PER INCH;
DIST	LOCATION OF NEXT COLUMN OF NODAL VALUES TO BE PRINTED;
FACT1	FACTOR FOR ADJUSTING VALUE OF DRAWDOWN PRINTED;
FACT2	FACTOR FOR ADJUSTING VALUE OF HEAD PRINTED;
K	ADJUSTED VALUE OF DRAWDOWN OR HEAD;
LA	LAYER FOR WHICH A MAP IS BEING PRINTED;
MESUR	NAME OF MAP LENGTH UNIT;
N	INDEX FOR SYMBOLS;
NA	INDICES FOR LOCATING X LABEL;
NC	NUMBER OF BLANKS BEFORE GRAPH;
NG	= 1, FOR DRAWDOWN MAP; = 2, FOR HEAD MAP;
N1	NUMBER OF LINES PER INCH;
N2	NUMBER OF CHARACTERS PER INCH;
N3	NUMBER OF CHARACTERS PER LINE;
N4	NUMBER OF LINES IN THE PLOT;
N8	MAXIMUM NUMBER OF CHARACTERS IN Y DIRECTION;
NXD	NUMBER OF INCHES IN THE X DIMENSION OF PLOT;
NYD	NUMBER OF INCHES IN THE Y DIMENSION OF PLOT;
PRNT	CONTAINS THE ARRANGEMENT OF SYMBOLS FOR EACH LINE;
SPACNG	CONTOUR INTERVAL (L);
SYM	VECTOR CONTAINING SYMBOLS USED IN THE PLOT;
TITLE	TITLE FOR PLOT;
VF1,VF2,VF3	VARIABLE FORMATS FOR CENTERING PLOT;
WIDTH	WIDTH OF MODEL (L);
XLABEL	LABEL FOR X AXIS;
XN	NUMBERS FOR X AXIS;

XN1	1 INCH/(N1*2);
XSCALE	MULTIPLICATION FACTOR TO CONVERT MODEL LENGTH UNIT TO UNIT USED IN X DIRECTION ON MAPS;
XSF	X SCALE FACTOR;
YDIM	LENGTH OF AQUIFER IN Y DIRECTION (L).
YLABEL	LABEL FOR Y AXIS;
YLEN	LOCATION OF NEXT VALUE IN THE COLUMN TO BE PRINTED;
YN	NUMBERS FOR Y AXIS;
YSCALE	MULTIPLICATION FACTOR TO CONVERT MODEL LENGTH UNIT TO UNIT USED IN Y DIRECTION ON MAPS;
YSF	Y SCALE FACTOR;
Z	LOCATION OF NEXT LINE TO BE PRINTED.

APPENDIX VI

Program Listing

```
C     -----------------------------------------------------------------
C     FINITE-DIFFERENCE MODEL FOR SIMULATION OF GROUND-WATER FLOW IN
C     THREE DIMENSIONS, SEPTEMBER, 1975 BY P.C. TRESCOTT, U. S. G. S.
C     WITH CONTRIBUTIONS TO MAIN, DATAI AND SOLVE BY S.P. LARSON
C     -----------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8YSTR
C
      DIMENSION Y(102000), L(25), HEADNG(33), NAME(42), INFT(2,2), IOFT(
     19,4), DUM(3)
C
      EQUIVALENCE (YSTR,Y(1))
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)
C
      DATA NAME/2*4H    ,4H   S,4HTART,4HING ,4HHEAD,4H    ,4H STO,4HRAG
     1E,4H COE,4HFFIC,4HIENT,2*4H    ,4H  TR,4HANSM,4HISSI,4HVITY,5*4H
     2   ,4H  TK,4H  HY,4HDRAU,4HLIC ,4HCOND,4HUCTI,4HVITY,2*4H    ,4HBOT
     3T,4HOM E,4HLEVA,4HTION,2*4H    ,4H   R,4HECHA,4HRGE ,4HRATE/
      DATA INFT/4H(20F,4H4.0),4H(8F1,4H0.4)/
      DATA IOFT/4H(1H0,4H,I2,,4H2X,2,4H0F6.,4H1/(5,4HX,20,4HF6.1,4H))  ,
     14H    ,4H(1H0,4H,I5,,4H14F9,4H.5/(,4H1H ,,4H5X,1,4H4F9.,4H5)) ,4H
     2    ,4H(1H0,4H,I5,,4H10EU,4H2.5/,4H(1H ,4H,5X,,4H10E1,4H2.5),4H)
     3,4H(1H0,4H,I5,,4H10E1,4H1.3/,4H(1H ,4H,5X,,4H10E1,4H1.3),4H)    /
C
      DEFINE FILE 2(8,1520,U,KKK)
C     .................................................................
C
C     ---READ TITLE, PROGRAM SIZE AND OPTIONS---
      READ (5,200) HEADNG
      WRITE (6,190) HEADNG
      READ (5,160) I0,J0,K0,ITMAX,NCH
      WRITE (6,180) I0,J0,K0,ITMAX,NCH
      READ (5,210) IDRAW,IHEAD,IFLO,IDK1,IDK2,IWATER,IQRE,IPU1,IPU2,ITK
      WRITE (6,220) IDRAW,IHEAD,IFLO,IDK1,IDK2,IWATER,IQRE,IPU1,IPU2,ITK
      IERR=0
C
C     ---COMPUTE DIMENSIONS FOR ARRAYS---
      J1=J0-1
      I1=I0-1
      K1=K0-1
      I2=I0-2
      J2=J0-2
      K2=K0-2
      IMAX=MAX0(I0,J0)
      NCD=MAX0(1,NCH)
      ITMX1=ITMAX+1
      ISIZ=I0*J0*K0
      IK1=I0*J0
      IK2=MAX0(IK1*K1,1)
      ISUM=2*ISIZ+1
```

```
      L(1)=1                                                            MAN0570
      DO 30 I=2,14                                                      MAN0580
      IF (I.NE.8) GO TO 20                                              MAN0590
      L(8)=ISUM                                                         MAN0600
      ISUM=ISUM+IK2                                                     MAN0610
      IF (IK2.EQ.1) GO TO 10                                            MAN0620
      IK=I0                                                             MAN0630
      JK=J0                                                             MAN0640
      K5=K1                                                             MAN0650
      GO TO 30                                                          MAN0660
   10 IK=1                                                              MAN0670
      JK=1                                                              MAN0680
      K5=1                                                              MAN0690
      GO TO 30                                                          MAN0700
   20 L(I)=ISUM                                                         MAN0710
      ISUM=ISUM+ISIZ                                                    MAN0720
   30 CONTINUE                                                          MAN0730
      L(15)=ISUM                                                        MAN0740
      ISUM=ISUM+J0                                                      MAN0750
      L(16)=ISUM                                                        MAN0760
      ISUM=ISUM+I0                                                      MAN0770
      L(17)=ISUM                                                        MAN0780
      ISUM=ISUM+K0                                                      MAN0790
      L(18)=ISUM                                                        MAN0800
      ISUM=ISUM+IMAX                                                    MAN0810
      L(19)=ISUM                                                        MAN0820
      ISUM=ISUM+K0*3                                                    MAN0830
      L(20)=ISUM                                                        MAN0840
      ISUM=ISUM+ITMX1                                                   MAN0850
      L(21)=ISUM                                                        MAN0860
      ISUM=ISUM+3*NCD                                                   MAN0870
      L(22)=ISUM                                                        MAN0880
      ISUM=ISUM+NCD                                                     MAN0890
      L(23)=ISUM                                                        MAN0900
      IF (IWATER.NE.ICHK(6)) GO TO 40                                   MAN0910
      ISUM=ISUM+IK1                                                     MAN0920
      L(24)=ISUM                                                        MAN0930
      ISUM=ISUM+IK1                                                     MAN0940
      IP=I0                                                             MAN0950
      JP=J0                                                             MAN0960
      GO TO 50                                                          MAN0970
   40 ISUM=ISUM+1                                                       MAN0980
      L(24)=ISUM                                                        MAN0990
      ISUM=ISUM+1                                                       MAN1000
      IP=1                                                              MAN1010
      JP=1                                                              MAN1020
   50 L(25)=ISUM                                                        MAN1030
      IF (IQRE.NE.ICHK(7)) GO TO 60                                     MAN1040
      ISUM=ISUM+IK1                                                     MAN1050
      IQ=I0                                                             MAN1060
      JQ=J0                                                             MAN1070
      GO TO 70                                                          MAN1080
   60 ISUM=ISUM+1                                                       MAN1090
      IQ=1                                                              MAN1100
      JQ=1                                                              MAN1110
   70 WRITE (6,170) ISUM                                                MAN1120
```

```
C                                                                       MAN1130
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)))                                                    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)))                MAN1230
      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)))                                                   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
C                                                                       MAN1320
C     ---START COMPUTATIONS---                                          MAN1330
C     ************************                                          MAN1340
C     ---READ AND WRITE DATA FOR GROUPS II AND III---                   MAN1350
      CALL DATAIN                                                       MAN1360
      IRN=1                                                             MAN1370
      NIJ=I0*J0                                                         MAN1380
      DO 80 K=1,K0                                                      MAN1390
      LOC=L(2)+(K-1)*NIJ                                                MAN1400
   80 CALL ARRAY(Y(LOC),INFT(1,2),IOFT(1,1),NAME(1),IRN,DUM)            MAN1410
      DO 90 K=1,K0                                                      MAN1420
      LOC=L(5)+(K-1)*NIJ                                                MAN1430
   90 CALL ARRAY(Y(LOC),INFT(1,1),IOFT(1,2),NAME(7),IRN,DUM)            MAN1440
      DO 100 K=1,K0                                                     MAN1450
      LOC=L(4)+(K-1)*NIJ                                                MAN1460
      L1=L(19)+K-1                                                      MAN1470
      L2=L(19)+K0+K-1                                                   MAN1480
      L3=L(19)+2*K0+K-1                                                 MAN1490
      CALL ARRAY(Y(LOC),INFT(1,1),IOFT(1,2),NAME(13),IRN,DUM)           MAN1500
      Y(L1)=DUM(1)                                                      MAN1510
      Y(L2)=DUM(2)                                                      MAN1520
      Y(L3)=DUM(3)                                                      MAN1530
  100 WRITE (6,230) K,Y(L1),Y(L2),Y(L3)                                 MAN1540
      IF (ITK.NE.ICHK(10)) GO TO 120                                    MAN1550
      DO 110 K=1,K1                                                     MAN1560
      LOC=L(8)+(K-1)*NIJ                                                MAN1570
  110 CALL ARRAY(Y(LOC),INFT(1,1),IOFT(1,3),NAME(19),IRN,DUM)           MAN1580
  120 IF (IWATER.NE.ICHK(6)) GO TO 130                                  MAN1590
      K = K0                                                            MAN1595
      CALL ARRAY(Y(L(23)),INFT(1,1),IOFT(1,4),NAME(25),IRN,DUM)         MAN1600
      CALL ARRAY(Y(L(24)),INFT(1,1),IOFT(1,1),NAME(31),IRN,DUM)         MAN1610
  130 IF (IQRE.EQ.ICHK(7)) CALL ARRAY(Y(L(25)),INFT(1,1),IOFT(1,4),NAME(MAN1620
     137),IRN,DUM)                                                      MAN1630
      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                                                         MAN1700
C                                                                       MAN1710
C     ---COMPUTE ITERATION PARAMETERS---                                MAN1720
      CALL ITER                                                         MAN1730
C                                                                       MAN1740
C     ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-MAN1750
  140 CALL NEWPER                                                       MAN1760
C                                                                       MAN1770
      KT=0                                                              MAN1780
      IFINAL=0                                                          MAN1790
C                                                                       MAN1800
C     ---START NEW TIME STEP COMPUTATIONS---                            MAN1810
  150 CALL NEWSTP                                                       MAN1820
C                                                                       MAN1830
C     ---START NEW ITERATION IF MAXIMUM NO. ITERATIONS NOT EXCEEDED---  MAN1840
      CALL NEWITA                                                       MAN1850
C                                                                       MAN1860
C     ---PRINT OUTPUT AT DESIGNATED TIME STEPS---                       MAN1870
      CALL OUTPUT                                                       MAN1880
C                                                                       MAN1890
C     ---LAST TIME STEP IN PUMPING PERIOD ?---                          MAN1900
      IF (IFINAL.NE.1) GO TO 150                                        MAN1910
C                                                                       MAN1920
C     ---CHECK FOR NEW PUMPING PERIOD---                                MAN1930
      IF (KP.LT.NPER) GO TO 140                                         MAN1940
C                                                                       MAN1950
      STOP                                                              MAN1960
C                                                                       MAN1970
C     ---FORMATS---                                                     MAN1980
C                                                                       MAN1990
C                                                                       MAN2000
C                                                                       MAN2010
  160 FORMAT (8I10)                                                     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)          MAN2060
  190 FORMAT ('1',33A4)                                                 MAN2070
  200 FORMAT (20A4)                                                     MAN2080
  210 FORMAT (16(A4,1X))                                                MAN2090
  220 FORMAT ('-SIMULATION OPTIONS:  ',11(A4,4X))                       MAN2100
  230 FORMAT (1H0,44X,'DIRECTIONAL TRANSMISSIVITY MULTIPLICATION FACTORSMAN2110
     1 FOR LAYER',I3,/,76X,'X =',G15.7/76X,'Y =',G15.7/76X,'Z =',G15.7) MAN2120
      END                                                               MAN2130-
```

```
      SUBROUTINE DATAI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACDAT0010
     1T,PERM,BOTTOM,QRE)                                                DAT0020
C     -----------------------------------------------------------------DAT0030
C     READ AND WRITE DATA                                               DAT0040
C     -----------------------------------------------------------------DAT0050
C                                                                       DAT0060
C     SPECIFICATIONS:                                                   DAT0070
      REAL *8PHI                                                        DAT0080
      REAL *8XLABEL,YLABEL,TITLE,XN1,MESUR                              DAT0090
C                                                                       DAT0100
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0DAT0110
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,DAT0120
     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
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        DAT0180
      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
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANKDAT0220
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),DAT0230
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                          DAT0240
      RETURN                                                            DAT0250
C     .................................................................DAT0260
C     ********************                                              DAT0270
      ENTRY DATAIN                                                      DAT0280
C     ********************                                              DAT0290
C                                                                       DAT0300
C     ---READ AND WRITE SCALAR PARAMETERS---                            DAT0310
      READ (5,330) NPER,KTH,ERR,LENGTH                                  DAT0320
      WRITE (6,340) NPER,KTH,ERR                                        DAT0330
      READ (5,460) XSCALE,YSCALE,DINCH,FACT1,(LEVEL1(I),I=1,9),FACT2,(LEDAT0340
     1VEL2(I),I=1,9),MESUR                                              DAT0350
      IF (XSCALE.NE.0.) WRITE (6,470) XSCALE,YSCALE,MESUR,MESUR,DINCH,FADAT0360
     1CT1,LEVEL1,FACT2,LEVEL2                                           DAT0370
C                                                                       DAT0380
C     ---READ CUMULATIVE MASS BALANCE PARAMETERS---                     DAT0390
      READ (5,450) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT0400
     1XT,FLXNT                                                          DAT0410
      IF (IDK1.EQ.ICHK(4)) GO TO 20                                     DAT0420
      IF (IPU1.NE.ICHK(8)) GO TO 50                                     DAT0430
C                                                                       DAT0440
C     ---READ INITIAL HEAD VALUES FROM CARDS---                         DAT0450
      DO 10 K=1,K0                                                      DAT0460
      DO 10 I=1,I0                                                      DAT0470
   10 READ (5,360) (PHI(I,J,K),J=1,J0)                                  DAT0480
      GO TO 30                                                          DAT0490
C                                                                       DAT0500
C     ---READ INITIAL HEAD AND MASS BALANCE PARAMETERS FROM DISK---     DAT0510
   20 READ (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT0520
     1XT,FLXNT                                                          DAT0530
      REWIND 4                                                          DAT0540
   30 WRITE (6,430) SUM                                                 DAT0550
      DO 40 K=1,K0                                                      DAT0560
```

```
      WRITE (6,440) K                                                   DAT0570
      DO 40 I=1,I0                                                      DAT0580
   40 WRITE (6,350) I,(PHI(I,J,K),J=1,J0)                               DAT0590
C                                                                       DAT0600
   50 DO 60 K=1,K0                                                      DAT0610
      DO 60 I=1,I0                                                      DAT0620
      DO 60 J=1,J0                                                      DAT0630
      WELL(I,J,K)=0.                                                    DAT0640
      TR(I,J,K)=0.                                                      DAT0650
      TC(I,J,K)=0.                                                      DAT0660
      IF (K.NE.K0) TK(I,J,K)=0.                                         DAT0670
   60 CONTINUE                                                          DAT0680
      RETURN                                                            DAT0690
C     *********************                                             DAT0700
      ENTRY ARRAY(A,INFT,IOFT,IN,IRN,TF)                                DAT0710
C     *********************                                             DAT0720
      READ (5,330) FAC,IVAR,IPRN,TF,IRECS,IRECD                         DAT0730
      IC=4*IRECS+2*IVAR+IPRN+1                                          DAT0740
      GO TO (70,70,90,90,120,120), IC                                   DAT0750
   70 DO 80 I=1,I0                                                      DAT0760
      DO 80 J=1,J0                                                      DAT0770
   80 A(I,J)=FAC                                                        DAT0780
      WRITE (6,280) IN,FAC,K                                            DAT0790
      GO TO 140                                                         DAT0800
   90 IF (IC.EQ.3) WRITE (6,290) IN,K                                   DAT0810
      DO 110 I=1,I0                                                     DAT0820
      READ (5,INFT) (A(I,J),J=1,J0)                                     DAT0830
      DO 100 J=1,J0                                                     DAT0840
  100 A(I,J)=A(I,J)*FAC                                                 DAT0850
  110 IF (IC.EQ.3) WRITE (6,IOFT) I,(A(I,J),J=1,J0)                     DAT0860
      GO TO 140                                                         DAT0870
  120 READ (2'IRN) A                                                    DAT0880
      IF (IC.EQ.6) GO TO 140                                            DAT0890
      WRITE (6,290) IN,K                                                DAT0900
      DO 130 I=1,I0                                                     DAT0910
  130 WRITE (6,IOFT) I,(A(I,J),J=1,J0)                                  DAT0920
  140 IF (IRECD.EQ.1) WRITE (2'IRN) A                                   DAT0930
      IRN=IRN+1                                                         DAT0940
      RETURN                                                            DAT0950
C     *********************                                             DAT0960
      ENTRY MDAT                                                        DAT0970
C     *********************                                             DAT0980
      DO 150 K=1,K0                                                     DAT0990
      DO 150 I=1,I0                                                     DAT1000
      DO 150 J=1,J0                                                     DAT1010
      IF (I.EQ.1.OR.I.EQ.I0.OR.J.EQ.1.OR.J.EQ.J0) T(I,J,K)=0.           DAT1020
      IF (IDK4.NE.ICHK(4).AND.IPU1.NE.ICHK(8)) PHI(I,J,K)=STRT(I,J,K)   DAT1030
      IF (K.NE.K0.OR.IWATER.NE.ICHK(6)) GO TO 150                       DAT1040
      IF (I.EQ.1.OR.I.EQ.I0.OR.J.EQ.1.OR.J.EQ.J0) PERM(I,J)=0.          DAT1050
  150 CONTINUE                                                          DAT1060
C     ............................ DELX ............................DAT1070
      READ (5,330) FAC,IVAR,IPRN                                        DAT1080
      IF (IVAR.EQ.1) READ (5,330) (DELX(J),J=1,J0)                      DAT1090
      DO 170 J=1,J0                                                     DAT1100
      IF (IVAR.NE.1) GO TO 160                                          DAT1110
      DELX(J)=DELX(J)*FAC                                               DAT1120
```

```
      GO TO 170
  160 DELX(J)=FAC
  170 CONTINUE
      IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,370) (DELX(J),J=1,J0)
      IF (IVAR.EQ.0) WRITE (6,300) FAC
C     ............................ DELY .............................
      READ (5,330) FAC,IVAR,IPRN
      IF (IVAR.EQ.1) READ (5,330) (DELY(I),I=1,I0)
      DO 190 I=1,I0
      IF (IVAR.NE.1) GO TO 180
      DELY(I)=DELY(I)*FAC
      GO TO 190
  180 DELY(I)=FAC
  190 CONTINUE
      IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,380) (DELY(I),I=1,I0)
      IF (IVAR.EQ.0) WRITE (6,310) FAC
C     ............................ DELZ .............................
      READ (5,330) FAC,IVAR,IPRN
      IF (IVAR.EQ.1) READ (5,330) (DELZ(K),K=1,K0)
      DO 210 K=1,K0
      IF (IVAR.NE.1) GO TO 200
      DELZ(K)=DELZ(K)*FAC
      GO TO 210
  200 DELZ(K)=FAC
  210 CONTINUE
      IF (IVAR.EQ.1.AND.IPRN.NE.1) WRITE (6,390) (DELZ(K),K=1,K0)
      IF (IVAR.EQ.0) WRITE (6,320) FAC
C
C     ---INITIALIZE VARIABLES---
      B=0.
      D=0.
      F=0.
      H=0.
      SU=0.
      Z=0.
      IF (XSCALE.NE.0.) CALL MAP
      RETURN
C     ...............................................................
C     ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-
C     *********************
      ENTRY NEWPER
C     *********************
C
      READ (5,330) KP,KPM1,NWEL,TMAX,NUMT,CDLT,DELT
C
C     ---COMPUTE ACTUAL DELT AND NUMT---
      DT=DELT/24.
      TM=0.0
      DO 220 I=1,NUMT
      DT=CDLT*DT
      TM=TM+DT
      IF (TM.GE.TMAX) GO TO 230
  220 CONTINUE
      GO TO 240
  230 DELT=TMAX/TM*DELT
      NUMT=I
```

```
  240 WRITE (6,400) KP,TMAX,NUMT,DELT,CDLT                              DAT1690
      DELT=DELT*3600.                                                   DAT1700
      TMAX=TMAX*86400.                                                  DAT1710
      SUMP=0.0                                                          DAT1720
C                                                                       DAT1730
C     ---READ AND WRITE WELL PUMPING RATES---                           DAT1740
      WRITE (6,410) NWEL                                                DAT1750
      IF (NWEL.EQ.0) GO TO 260                                          DAT1760
      DO 245 K = 1,K0                                                   DAT1761
      DO 245 I = 1,I0                                                   DAT1762
      DO 245 J = 1,J0                                                   DAT1763
  245 WELL(I,J,K) = 0.0                                                 DAT1764
      DO 250 II=1,NWEL                                                  DAT1770
      READ (5,330) K,I,J,WELL(I,J,K)                                    DAT1780
      WRITE (6,420) K,I,J,WELL(I,J,K)                                   DAT1790
  250 WELL(I,J,K)=WELL(I,J,K)/(DELX(J)*DELY(I))                         DAT1800
  260 RETURN                                                            DAT1810
C                                                                       DAT1820
C     ---FORMATS---                                                     DAT1830
C                                                                       DAT1840
C                                                                       DAT1850
C                                                                       DAT1860
  280 FORMAT (1H0,52X,6A4,' =',G15.7,' FOR LAYER',I3)                   DAT1870
  290 FORMAT (1H1,45X,6A4,' MATRIX, LAYER',I3/46X,41('-'))             DAT1880
  300 FORMAT ('0',72X,'DELX =',G15.7)                                   DAT1890
  310 FORMAT ('0',72X,'DELY =',G15.7)                                   DAT1900
  320 FORMAT ('0',72X,'DELZ =',G15.7)                                   DAT1910
  330 FORMAT (8G10.0)                                                   DAT1920
  340 FORMAT ('0',51X,'NUMBER OF PUMPING PERIODS =',I5/49X,'TIME STEPS BDAT1930
     1ETWEEN PRINTOUTS =',I5//51X,'ERROR CRITERIA FOR CLOSURE =',G15.7/)DAT1940
  350 FORMAT ('0',I2,2X,20F6.1/(5X,20F6.1))                             DAT1950
  360 FORMAT (8F10.4)                                                   DAT1960
  370 FORMAT (1H1,46X,40HGRID SPACING IN PROTOTYPE IN X DIRECTION/47X,40DAT1970
     1('-')//('0',12F10.0))                                             DAT1980
  380 FORMAT (1H-,46X,40HGRID SPACING IN PROTOTYPE IN Y DIRECTION/47X,40DAT1990
     1('-')//('0',12F10.0))                                             DAT2000
  390 FORMAT (1H-,46X,40HGRID SPACING IN PROTOTYPE IN Z DIRECTION/47X,40DAT2010
     1('-')//('0',12F10.0))                                             DAT2020
  400 FORMAT ('-',50X,'PUMPING PERIOD NO.',I4,':',F10.2,' DAYS'/51X,38('DAT2030
     1-')//53X,'NUMBER OF TIME STEPS=',I6//59X,'DELT IN HOURS =',F10.3//DAT2040
     253X,'MULTIPLIER FOR DELT =',F10.3)                                DAT2050
  410 FORMAT ('-',63X,I4,' WELLS'/65X,9('-')//50X,'K',9X,'I',9X,'J    PUDAT2060
     1MPING RATE'/)                                                     DAT2070
  420 FORMAT (41X,3I10,2F13.2)                                          DAT2080
  430 FORMAT ('-',40X,' CONTINUATION - HEAD AFTER ',G20.7,' SEC PUMPING DAT2090
     1'/42X,58('-'))                                                    DAT2100
  440 FORMAT ('1',55X,'INITIAL  HEAD MATRIX, LAYER',I3/56X,30('-'))     DAT2110
  450 FORMAT (4G20.10)                                                  DAT2120
  460 FORMAT (3G10.0,2(G10.0,9I1,1X),A8)                                DAT2130
  470 FORMAT ('0',30X,'ON ALPHAMERIC MAP:'/40X,'MULTIPLICATION FACTOR FODAT2140
     1R X DIMENSION =',G15.7/40X,'MULTIPLICATION FACTOR FOR Y DIMENSION DAT2150
     2=',G15.7/55X,'MAP SCALE IN UNITS OF  ',A11/50X,'NUMBER OF ',A8,' PDAT2160
     3ER INCH =',G15.7/43X,'MULTIPLICATION FACTOR FOR DRAWDOWN =',G15.7,DAT2170
     4' PRINTED FOR LAYERS',9I2/47X,'MULTIPLICATION FACTOR FOR HEAD =',GDAT2180
     515.7,' PRINTED FOR LAYERS',9I2)                                   DAT2190
      END                                                               DAT2200-
```

```
      SUBROUTINE STEP(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACT
     1,DDN,TEST3)
C     ------------------------------------------------------------------
C     INITIALIZE DATA FOR A NEW TIME STEP AND PRINT RESULTS
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI
      REAL *8XLABEL,YLABEL,TITLE,XN1,MESUR
C
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(K0,3), DDN(IMAX), TEST3
     3(ITMX1), ITTO(50)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANK
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2
      RETURN
C     ..................................................................
C     ********************
      ENTRY NEWSTP
C     ********************
      KT=KT+1
      IT=0
      DO 10 K=1,K0
      DO 10 I=1,I0
      DO 10 J=1,J0
   10 OLD(I,J,K)=PHI(I,J,K)
      DELT=CDLT*DELT
      SUM=SUM+DELT
      SUMP=SUMP+DELT
      DAYSP=SUMP/86400.
      YRSP=DAYSP/365.
      HRS=SUM/3600.
      SMIN=HRS*60.
      DAYS=HRS/24.
      YRS=DAYS/365.
      RETURN
C
C     ---PRINT OUTPUT AT DESIGNATED TIME STEPS---
C     *********************
      ENTRY OUTPUT
C     *********************
      IF (KT.EQ.NUMT) IFINAL=1
      ITTO(KT)=IT
      IF (IT.LE.ITMAX) GO TO 20
      IT=IT-1
      ITTO(KT)=IT
      IERR=2
```

```
C                                                                       STP 570
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                                    STP 600
      IF (IPU2.EQ.ICHK(9)) WRITE (7,230) SUM,SUMP,PUMPT,CFLUXT,QRET,CHSTSTP 610
     1,CHDT,FLUXT,STORT,ETFLXT,FLXNT                                    STP 620
C                                                                       STP 630
   20 IF (IFLO.EQ.ICHK(3)) CALL CHECK                                   STP 640
      IF (IERR.EQ.2) GO TO 30                                           STP 650
      IF (MOD(KT,KTH).NE.0.AND.IFINAL.NE.1) RETURN                      STP 660
   30 WRITE (6,210) KT,DELT,SUM,SMIN,HRS,DAYS,YRS,DAYSP,YRSP            STP 670
      IF (IFLO.EQ.ICHK(3)) CALL CWRITE                                  STP 680
      IT=IT+1                                                           STP 690
      WRITE (6,180) (TEST3(J),J=1,IT)                                   STP 700
      I3=1
      I5=0
  352 I5=I5+40
      I4=MIN0(KT,I5)
      WRITE (6,240) (I,I=I3,I4)                                         STP 710
      WRITE (6,260)                                                     STP 720
      WRITE (6,250) (ITTO(I),I=I3,I4)                                   STP 730
      WRITE (6,260)                                                     STP 740
      IF(KT.LE.I5) GO TO 353
      I3=I3+40
      GO TO 352
C                                                                       STP 750
C     ---PRINT MAPS---                                                  STP 760
  353 IF (XSCALE.EQ.0.) GO TO 70                                        STP 770
      IF (FACT1.EQ.0.) GO TO 50                                         STP 780
      DO 40 IA=1,9                                                      STP 790
      II=LEVEL1(IA)                                                     STP 800
      IF (II.EQ.0) GO TO 50                                             STP 810
   40 CALL PRNTA(1,II)                                                  STP 820
   50 IF (FACT2.EQ.0.) GO TO 70                                         STP 830
      DO 60 IA=1,9                                                      STP 840
      II=LEVEL2(IA)                                                     STP 850
      IF (II.EQ.0) GO TO 70                                             STP 860
   60 CALL PRNTA(2,II)                                                  STP 870
   70 IF (IDRAW.NE.ICHK(1)) GO TO 100                                   STP 880
C                                                                       STP 890
C     ---PRINT DRAWDOWN---                                              STP 900
      DO 90 K=1,K0                                                      STP 910
      WRITE (6,200) K                                                   STP 920
      DO 90 I=1,I0                                                      STP 930
      DO 80 J=1,J0                                                      STP 940
   80 DDN(J)=STRT(I,J,K)-PHI(I,J,K)                                     STP 950
   90 WRITE (6,170) I,(DDN(J),J=1,J0)                                   STP 960
  100 IF (IHEAD.NE.ICHK(2)) GO TO 120                                   STP 970
C                                                                       STP 980
C     ---PRINT HEAD MATRIX---                                           STP 990
      DO 110 K=1,K0                                                     STP1000
      WRITE (6,190) K                                                   STP1010
      DO 110 I=1,I0                                                     STP1020
  110 WRITE (6,170) I,(PHI(I,J,K),J=1,J0)                               STP1030
C                                                                       STP1040
C     ---WRITE ON DISK---                                               STP1050
```

```
  120 IF (IERR.EQ.2) GO TO 130                                          STP1060
      IF (KP.LT.NPER.OR.IFINAL.NE.1) RETURN                             STP1070
      IF (IDK2.EQ.ICHK(5)) WRITE (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHSTSTP1080
     1,CHDT,FLUXT,STORT,ETFLXT,FLXNT                                    STP1090
C                                                                       STP1100
C     ---PUNCHED OUTPUT---                                              STP1110
  130 IF (IPU2.NE.ICHK(9)) GO TO 160                                    STP1120
      IF (IERR.EQ.2) GO TO 140                                          STP1130
      WRITE (7,230) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFSTP1140
     1LXT,FLXNT                                                         STP1150
  140 DO 150 K=1,K0                                                     STP1160
  150 WRITE (7,220) ((PHI(I,J,K),J=1,J0),I=1,I0)                        STP1170
  160 IF (IERR.EQ.2) STOP                                               STP1180
      RETURN                                                            STP1190
C                                                                       STP1200
C     ---FORMATS---                                                     STP1210
C                                                                       STP1220
C                                                                       STP1230
C                                                                       STP1240
  170 FORMAT ('0',I4,18F7.2/(5X,18F7.2))                                STP1250
  180 FORMAT ('0MAXIMUM HEAD CHANGE FOR EACH ITERATION:'/' ',39('-')/('0STP1260
     1',10F12.4))                                                       STP1270
  190 FORMAT ('1',55X,'HEAD MATRIX, LAYER',I3/56X,21('-'))              STP1280
  200 FORMAT ('1',55X,'   DRAWDOWN, LAYER',I3/59X,18('-'))              STP1290
  210 FORMAT (1H1,44X,57('-')/45X,'|',14X,'TIME STEP NUMBER =',I9,14X,'|STP1300
     1'/45X,57('-')//50X,29HSIZE OF TIME STEP IN SECONDS=,F14.2//55X,'TOSTP1310
     2TAL SIMULATION TIME IN SECONDS=',F14.2/80X,8HMINUTES=,F14.2/82X,6HSTP1320
     3HOURS=,F14.2/83X,5HDAYS=,F14.2/82X,'YEARS=',F14.2///45X,'DURATION STP1330
     4OF CURRENT PUMPING PERIOD IN DAYS=',F14.2/82X,'YEARS=',F14.2//)  STP1340
  220 FORMAT (10F8.2)                                                   STP1350
  230 FORMAT (4G20.10)                                                  STP1360
  240 FORMAT ('0TIME STEP :',40I3)                                      STP1370
  250 FORMAT ('0ITERATIONS:',40I3)                                      STP1380
  260 FORMAT (' ',10('-'))                                              STP1390
      END                                                               STP1400-
```

```
      SUBROUTINE SOLVE(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FAC
     1T,EL,FL,GL,V,XI,TEST3,QRE)
C     ------------------------------------------------------------------
C     SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI,RHO,B,D,F,H,Z,SU,RHOP,W,WMIN,RHO1,RHO2,RHO3,XPART,YPART
     1,ZPART,DMIN1,WMAX,XT,YT,ZT,DABS,DMAX1,DEN,TXM,TYM,TZM
      REAL *8E,AL,BL,CL,A,C,G,WU,TU,U,DL,RES,SUPH,GLXI,ZPHI
C
      DIMENSION PHI(1), STRT(1), OLD(1), T(1), S(1), TR(1), TC(1), TK(1)
     1, WELL(1), DELX(1), DELY(1), DELZ(1), FACT(K0,3), RHOP(20), TEST3(
     21), EL(1), FL(1), GL(1), V(1), XI(1), QRE(1)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)
      RETURN
C     ..................................................................
C     *********************
      ENTRY ITER
C     *********************
C     ---COMPUTE AND PRINT ITERATION PARAMETERS---
      WRITE (6,240)
      WMIN=1.D0
      DELT=1.
      P2=LENGTH-1
      NT=I0*J0*K0
      NIJ=I0*J0
      XT=3.141593**2/(2.*J2*J2)
      YT=3.141593**2/(2.*I2*I2)
      ZT=3.141593**2/(2.*K0*K0)
      RHO1=0.D0
      RHO2=0.D0
      RHO3=0.D0
      DO 40 K=1,K0
      DO 40 I=2,I1
      DO 40 J=2,J1
      N=I+(J-1)*I0+(K-1)*NIJ
      IF (T(N).EQ.0.) GO TO 40
      D=TR(N-I0)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      SU=0,D0
      Z=0.D0
      IF (K.NE.1) Z=TK(N-NIJ)/DELZ(K)
      IF (K.NE.K0) SU=TK(N)/DELZ(K)
      RHO=S(N)/DELT
      QR=0.
      IF (K.NE.K0) GO TO 10
      IF (IQRE.EQ.ICHK(7)) QR=QRE(I+(J-1)*I0)
   10 CONTINUE
```

```
      TXM=DMAX1(D,F)
      TYM=DMAX1(B,H)
      TZM=DMAX1(SU,Z)
      DEN=DMIN1(D,F)
      IF (DEN.EQ.0.D0) DEN=TXM
      IF (DEN.EQ.0.D0) GO TO 20
      RHO1=DMAX1(RHO1,TYM/DEN)
   20 DEN=DMIN1(B,H)
      IF (DEN.EQ.0.D0) DEN=TYM
      IF (DEN.EQ.0.D0) GO TO 30
      RHO2=DMAX1(RHO2,TXM/DEN)
   30 DEN=DMIN1(SU,Z)
      IF (DEN.EQ.0.D0) DEN=TZM
      IF (DEN.EQ.0.D0) GO TO 40
      RHO3=DMAX1(RHO3,TXM/DEN)
   40 CONTINUE
      XPART=XT/(1.D0+RHO1)
      YPART=YT/(1.D0+RHO2)
      ZPART=ZT/(1.D0+RHO3)
      WMIN=DMIN1(WMIN,XPART,YPART,ZPART)
      WMAX=1.D0-WMIN
      PJ=-1.
      DO 50 I=1,LENGTH
      PJ=PJ+1.
   50 RHOP(I)=1.D0-(1.D0-WMAX)**(PJ/P2)
      WRITE (6,230) LENGTH,(RHOP(J),J=1,LENGTH)
      RETURN
C     ..................................................................
C
C     ---INITIALIZE DATA FOR A NEW ITERATION---
   60 IT=IT+1
      IF (IT.LE.ITMAX) GO TO 70
      WRITE (6,220)
      CALL OUTPUT
   70 IF (MOD(IT,LENGTH)) 80,80,90
C     ********************
      ENTRY NEWITA
C     ********************
   80 NTH=0
   90 NTH=NTH+1
      W=RHOP(NTH)
      TEST3(IT+1)=0.
      TEST=0.0
      BIG=0.
      DO 100 I=1,NT
      EL(I)=0.
      FL(I)=0.
      GL(I)=0.
      V(I)=0.
  100 XI(I)=0.
C
C     ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS FOR UPPER
C     HYDROLOGIC UNIT WHEN IT IS UNCONFINED---
      IF (IWATER.NE.ICHK(6)) GO TO 110
      CALL TRANS(0)
C
```

```
C     ---CHOOSE SIP NORMAL OR REVERSE ALGORITHM---
  110 IF (MOD(IT,2)) 120,120,170
  120 DO 150 K=1,K0
      DO 150 I=2,I1
      DO 150 J=2,J1
      N=I+(J-1)*I0+(K-1)*NIJ
      NIA=N+1
      NIB=N-1
      NJA=N+I0
      NJB=N-I0
      NKA=N+NIJ
      NKB=N-NIJ
C
C     ---SKIP COMPUTATIONS IF NODE OUTSIDE MODEL---
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 150
C
C     ---COMPUTE COEFFICIENTS---
      D=TR(NJB)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(NIB)/DELY(I)
      H=TC(N)/DELY(I)
      SU=0.D0
      Z=0.D0
      IF (K.NE.1) Z=TK(NKB)/DELZ(K)
      IF (K.NE.K0) SU=TK(N)/DELZ(K)
      RHO=S(N)/DELT
      QR=0.
      IF (K.NE.K0) GO TO 130
      IF (IQRE.EQ.ICHK(7)) QR=QRE(I+(J-1)*I0)
C
C     ---SIP NORMAL ALGORITHM---
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
  130 E=-B-D-F-H-SU-Z-RHO
      BL=B/(1.+W*(EL(NIB)+GL(NIB)))
      CL=D/(1.+W*(FL(NJB)+GL(NJB)))
      C=BL*EL(NIB)
      G=CL*FL(NJB)
      WU=CL*GL(NJB)
      U=BL*GL(NIB)
      IF (K.EQ.1) GO TO 140
      AL=Z/(1.+W*(EL(NKB)+FL(NKB)))
      A=AL*EL(NKB)
      TU=AL*FL(NKB)
      DL=E+W*(A+C+G+WU+TU+U)-CL*EL(NJB)-BL*FL(NIB)-AL*GL(NKB)
      EL(N)=(F-W*(A+C))/DL
      FL(N)=(H-W*(G+TU))/DL
      GL(N)=(SU-W*(WU+U))/DL
      SUPH=0.D0
      IF (K.NE.K0) SUPH=SU*PHI(NKA)
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-Z*P
     1HI(NKB)-WELL(N)-RHO*OLD(N)-QR
      V(N)=(RES-AL*V(NKB)-BL*V(NIB)-CL*V(NJB))/DL
      GO TO 150
  140 DL=E+W*(C+G+WU+U)-CL*EL(NJB)-BL*FL(NIB)
      EL(N)=(F-W*C)/DL
      FL(N)=(H-W*G)/DL
```

```
      GL(N)=(SU-W*(WU+U))/DL                                            SP31690
      SUPH=0.D0                                                         SP31700
      IF (K.NE.K0) SUPH=SU*PHI(NKA)                                     SP31710
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SUPH-WELSP31720
     1L(N)-RHO*OLD(N)-QR                                                SP31730
      V(N)=(RES-BL*V(NIB)-CL*V(NJB))/DL                                 SP31740
  150 CONTINUE                                                          SP31750
C                                                                       SP31760
C     ---BACK SUBSTITUTE FOR VECTOR XI---                               SP31770
      DO 160 K=1,K0                                                     SP31780
      K3=K0-K+1                                                         SP31790
      DO 160 I=1,I2                                                     SP31800
      I3=I0-I                                                           SP31810
      DO 160 J=1,J2                                                     SP31820
      J3=J0-J                                                           SP31830
      N=I3+(J3-1)*I0+(K3-1)*NIJ+I-I                                     SP31840
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 160                           SP31850
      GLXI=0.D0                                                         SP31860
      IF (K3.NE.K0) GLXI=GL(N)*XI(N+NIJ)                                SP31870
      XI(N)=V(N)-EL(N)*XI(N+I0)-FL(N)*XI(N+1)-GLXI                      SP31880
C                                                                       SP31890
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERIA---           SP31900
      TCHK=ABS(XI(N))                                                   SP31910
      IF (TCHK.GT.BIG) BIG=TCHK                                         SP31920
      PHI(N)=PHI(N)+XI(N)                                               SP31930
  160 CONTINUE                                                          SP31940
      IF (BIG.GT.ERR) TEST=1.                                           SP31950
      TEST3(IT+1)=BIG                                                   SP31960
      IF (TEST.EQ.0.) RETURN                                            SP31970
      GO TO 60                                                          SP31980
C     ..................................................................SP31990
  170 DO 200 KK=1,K0                                                    SP32000
      K=K0-KK+1                                                         SP32010
      DO 200 II=1,I2                                                    SP32020
      I=I0-II                                                           SP32030
      DO 200 J=2,J1                                                     SP32040
      N=I+(J-1)*I0+(K-1)*NIJ                                            SP32050
      NIA=N+1                                                           SP32060
      NIB=N-1                                                           SP32070
      NJA=N+I0                                                          SP32080
      NJB=N-I0                                                          SP32090
      NKA=N+NIJ                                                         SP32100
      NKB=N-NIJ                                                         SP32110
C                                                                       SP32120
C     ---SKIP COMPUTATIONS IF NODE OUTSIDE AQUIFER---                   SP32130
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 200                           SP32140
C                                                                       SP32150
C     ---COMPUTE COEFFICIENTS---                                        SP32160
      D=TR(NJB)/DELX(J)                                                 SP32170
      F=TR(N)/DELX(J)                                                   SP32180
      B=TC(NIB)/DELY(I)                                                 SP32190
      H=TC(N)/DELY(I)                                                   SP32200
      SU=0.D0                                                           SP32210
      Z=0.D0                                                            SP32220
      IF (K.NE.1) Z=TK(NKB)/DELZ(K)                                     SP32230
      IF (K.NE.K0) SU=TK(N)/DELZ(K)                                     SP32240
```

```
      RHO=S(N)/DELT
      QR=0.
      IF (K.NE.K0) GO TO 180
      IF (IQRE.EQ.ICHK(7)) QR=QRE(I+(J-1)*I0)
C
C     ---SIP REVERSE ALGORITHM---
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
  180 E=-B-D-F-H-SU-Z-RHO
      BL=H/(1.+W*(EL(NIA)+GL(NIA)))
      CL=D/(1.+W*(FL(NJB)+GL(NJB)))
      C=BL*EL(NIA)
      G=CL*FL(NJB)
      WU=CL*GL(NJB)
      U=BL*GL(NIA)
      IF (K.EQ.K0) GO TO 190
      AL=SU/(1.+W*(EL(NKA)+FL(NKA)))
      A=AL*EL(NKA)
      TU=AL*FL(NKA)
      DL=E+W*(C+G+A+WU+TU+U)-AL*GL(NKA)-BL*FL(NIA)-CL*EL(NJB)
      EL(N)=(F-W*(C+A))/DL
      FL(N)=(B-W*(G+TU))/DL
      GL(N)=(Z-W*(WU+U))/DL
      ZPHI=0.D0
      IF (K.NE.1) ZPHI=Z*PHI(NKB)
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-SU*PHI(N
     1KA)-ZPHI-WELL(N)-RHO*OLD(N)-QR
      V(N)=(RES-AL*V(NKA)-BL*V(NIA)-CL*V(NJB))/DL
      GO TO 200
  190 DL=E+W*(C+G+WU+U)-BL*FL(NIA)-CL*EL(NJB)
      EL(N)=(F-W*C)/DL
      FL(N)=(B-W*G)/DL
      GL(N)=(Z-W*(WU+U))/DL
      ZPHI=0.D0
      IF (K.NE.1) ZPHI=Z*PHI(NKB)
      RES=-B*PHI(NIB)-D*PHI(NJB)-E*PHI(N)-F*PHI(NJA)-H*PHI(NIA)-ZPHI-WEL
     1L(N)-RHO*OLD(N)-QR
      V(N)=(RES-BL*V(NIA)-CL*V(NJB))/DL
  200 CONTINUE
C
C     ---BACK SUBSTITUTE FOR VECTOR XI---
      DO 210 K=1,K0
      DO 210 I=2,I1
      DO 210 J=1,J2
      J3=J0-J
      N=I+(J3-1)*I0+(K-1)*NIJ
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 210
      GLXI=0.D0
      IF (K.NE.1) GLXI=GL(N)*XI(N-NIJ)
      XI(N)=V(N)-EL(N)*XI(N+I0)-FL(N)*XI(N-1)-GLXI
C
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERIA---
      TCHK=ABS(XI(N))
      IF (TCHK.GT.BIG) BIG=TCHK
      PHI(N)=PHI(N)+XI(N)
  210 CONTINUE
      IF (BIG.GT.ERR) TEST=1.
```

```
      TEST3(IT+1)=BIG                                                   SP32810
      IF (TEST.EQ.0.) RETURN                                            SP32820
      GO TO 60                                                          SP32830
C     ..................................................................SP32840
C                                                                       SP32850
C     ---FORMATS---                                                     SP32860
C                                                                       SP32870
C                                                                       SP32880
C                                                                       SP32890
  220 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))   SP32900
  230 FORMAT (///1H0,I5,22H ITERATION PARAMETERS:,6E15.7/(/28X,6E15.7/))SP32910
  240 FORMAT ('-',44X,'SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE'/45X,SP32920
     143('_'))                                                          SP32930
      END                                                               SP32940
```

```
      SUBROUTINE COEF (PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FACT
     1,PERM,BOTTOM,QRE)
C     ------------------------------------------------------------------
C     COMPUTE COEFFICIENTS
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI
C
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(K0,3), PERM(IP,JP), BOT
     3TOM(IP,JP), QRE(IQ,JQ)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,NC
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)
      RETURN
C     ..................................................................
C     ---COMPUTE TRANSMISSIVITY FOR UPPER HYDROLOGIC UNIT WHEN
C     IT IS UNCONFINED---
C     *********************
      ENTRY TRANS(N3)
C     *********************
      DO 10 I=2,I1
      DO 10 J=2,J1
      IF (PERM(I,J).EQ.0.) GO TO 10
      T(I,J,K0)=PERM(I,J)*(PHI(I,J,K0)-BOTTOM(I,J))
      IF (T(I,J,K0).GT.0.) GO TO 10
      IF (WELL(I,J,K0).LT.0.) WRITE (6,60) I,J,K0
      IF (WELL(I,J,K0).GE.0.) WRITE (6,70) I,J,K0
      PERM(I,J)=0.
      T(I,J,K0)=0.
      TR(I,J-1,K0)=0.
      TR(I,J,K0)=0.
      TC(I,J,K0)=0.
      TC(I-1,J,K0)=0.
      IF (K0.NE.1) TK(I,J,K1)=0.
      PHI(I,J,K0)=1.D30
   10 CONTINUE
      IF (N3.EQ.1) RETURN
      N1=K0
      N2=K0
      N4=K1
      GO TO 20
C     ---COMPUTE T COEFFICIENTS---
C     *********************
      ENTRY TCOF
C     *********************
      N1=1
      N2=K0
      N4=1
   20 DO 40 K=N1,N2
      DO 40 I=1,I1
```

```
      DO 40 J=1,J1
      IF (T(I,J,K).EQ.0.) GO TO 40
      IF (T(I,J+1,K).EQ.0.) GO TO 30
      TR(I,J,K)=(2.*T(I,J+1,K)*T(I,J,K))/(T(I,J,K)*DELX(J+1)+T(I,J+1,K)*
     1DELX(J))*FACT(K,1)
   30 IF (T(I+1,J,K).EQ.0.) GO TO 40
      TC(I,J,K)=(2.*T(I+1,J,K)*T(I,J,K))/(T(I,J,K)*DELY(I+1)+T(I+1,J,K)*
     1DELY(I))*FACT(K,2)
   40 CONTINUE
      IF(KO.EQ.1.OR.ITK.EQ.ICHK(10).OR.N3.EQ.0) RETURN
      DO 50 K=N4,K1
      DO 50 I=2,I1
      DO 50 J=2,J1
      IF (T(I,J,K+1).EQ.0.) GO TO 50
      T1=T(I,J,K)*FACT(K,3)
      T2=T(I,J,K+1)*FACT(K+1,3)
      TK(I,J,K)=(2.*T2*T1)/(T1*DELZ(K+1)+T2*DELZ(K))
   50 CONTINUE
      RETURN
C
C
   60 FORMAT ('-',20('*'),'WELL',2I3,' IN LAYER',I3,' GOES DRY',20('*'))
   70 FORMAT ('-',20('*'),'NODE',2I3,' IN LAYER',I3,' GOES DRY',20('*'))
      END
```

```
      SUBROUTINE CHECKI(PHI,STRT,OLD,T,S,TR,TC,TK,WELL,DELX,DELY,DELZ,FA
     1CT,JFLO,FLOW,QRE)
C     ------------------------------------------------------------------
C     COMPUTE A VOLUMETRIC BALANCE
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI
C
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), OLD(I0,J0,K0), T(I0,J0,K0
     1), S(I0,J0,K0), TR(I0,J0,K0), TC(I0,J0,K0), TK(IK,JK,K5), WELL(I0,
     2J0,K0), DELX(J0), DELY(I0), DELZ(K0), FACT(K0,3), JFLO(NCH,3), FLO
     3W(NCH), QRE(IQ,JQ), IQQ(40,38)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK
      COMMON /SPARAM/ TMAX,CDLT,DELT,ERR,TEST,SUM,SUMP,QR
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT
      RETURN
C     ..................................................................
C     ********************
      ENTRY CHECK
C     ********************
C     ---INITIALIZE VARIABLES---
      PUMP=0.
      STOR=0.
      FLUXS=0.0
      CHD1=0.0
      CHD2=0.0
      QREFLX=0.
      CFLUX=0.
      FLUX=0.
      ETFLUX=0.
      FLXN=0.0
      II=0
C     ..................................................................
C
C     ---COMPUTE RATES,STORAGE AND PUMPAGE FOR THIS STEP---
      DO 220 K=1,K0
      DO 220 I=2,I1
      DO 220 J=2,J1
      IF (T(I,J,K).EQ.0.) GO TO 220
      AREA=DELX(J)*DELY(I)
      IF (S(I,J,K).GE.0.) GO TO 180
C
C     ---COMPUTE FLOW RATES TO AND FROM CONSTANT HEAD BOUNDARIES---
      II=II+1
      FLOW(II)=0.
      JFLO(II,1)=K
      JFLO(II,2)=I
      JFLO(II,3)=J
      IF (S(I,J-1,K).LT.0..OR.T(I,J-1,K).EQ.0.) GO TO 30
      X=(PHI(I,J,K)-PHI(I,J-1,K))*TR(I,J-1,K)*DELY(I)
      FLOW(II)=FLOW(II)+X
```

```
      IF (X) 10,30,20
   10 CHD1=CHD1+X
      GO TO 30
   20 CHD2=CHD2+X
   30 IF (S(I,J+1,K).LT.0..OR.T(I,J+1,K).EQ.0.) GO TO 60
      X=(PHI(I,J,K)-PHI(I,J+1,K))*DELY(I)*TR(I,J,K)
      FLOW(II)=FLOW(II)+X
      IF (X) 40,60,50
   40 CHD1=CHD1+X
      GO TO 60
   50 CHD2=CHD2+X
   60 IF (K.EQ.1) GO TO 90
      IF (S(I,J,K-1).LT.0..OR.T(I,J,K-1).EQ.0.) GO TO 90
      X=(PHI(I,J,K)-PHI(I,J,K-1))*TK(I,J,K-1)*AREA*2./(DELZ(K)+DELZ(K-1)
     1)
      FLOW(II)=FLOW(II)+X
      IF (X) 70,90,80
   70 CHD1=CHD1+X
      GO TO 90
   80 CHD2=CHD2+X
   90 IF (K.EQ.KO) GO TO 120
      IF (S(I,J,K+1).LT.0..OR.T(I,J,K+1).EQ.0.) GO TO 120
      X=(PHI(I,J,K)-PHI(I,J,K+1))*TK(I,J,K)*AREA*2./(DELZ(K)+DELZ(K+1))
      FLOW(II)=FLOW(II)+X
      IF (X) 100,120,110
  100 CHD1=CHD1+X
      GO TO 120
  110 CHD2=CHD2+X
  120 IF (S(I-1,J,K).LT.0..OR.T(I-1,J,K).EQ.0.) GO TO 150
      X=(PHI(I,J,K)-PHI(I-1,J,K))*TC(I-1,J,K)*DELX(J)
      FLOW(II)=FLOW(II)+X
      IF (X) 130,150,140
  130 CHD1=CHD1+X
      GO TO 150
  140 CHD2=CHD2+X
  150 IF (S(I+1,J,K).LT.0..OR.T(I+1,J,K).EQ.0.) GO TO 220
      X=(PHI(I,J,K)-PHI(I+1,J,K))*TC(I,J,K)*DELX(J)
      FLOW(II)=FLOW(II)+X
      IF (X) 160,220,170
  160 CHD1=CHD1+X
      GO TO 220
  170 CHD2=CHD2+X
      GO TO 220
C
C     ---RECHARGE AND WELLS---
  180 IF (K.EQ.KO.AND.IQRE.EQ.ICHK(7)) QREFLX=QREFLX+QRE(I,J)*AREA
      IF (WELL(I,J,K)) 190,210,200
  190 PUMP=PUMP+WELL(I,J,K)*AREA
      GO TO 210
  200 CFLUX=CFLUX+WELL(I,J,K)*AREA
C
C     ---COMPUTE VOLUME FROM STORAGE---
  210 STOR=STOR+S(I,J,K)*(OLD(I,J,K)-PHI(I,J,K))*AREA
  220 CONTINUE
C     ..................................................................
C
```

```
C     ---COMPUTE CUMULATIVE VOLUMES, TOTALS, AND DIFFERENCES---
      FLXPT=0.0
      STORT=STORT+STOR
      STOR=STOR/DELT
      QRET=QRET+QREFLX*DELT
      CHDT=CHDT-CHD1*DELT
      CHST=CHST+CHD2*DELT
      PUMPT=PUMPT-PUMP*DELT
      CFLUXT=CFLUXT+CFLUX*DELT
      TOTL1=STORT+QRET+CFLUXT+CHST+FLXPT
      TOTL2=CHDT+PUMPT+ETFLXT+FLXNT
      SUMR=QREFLX+CFLUX+CHD2+CHD1+PUMP+ETFLUX+FLUXS+STOR
      DIFF=TOTL2-TOTL1
      PERCNT=0.0
      IF (TOTL2.EQ.0.) GO TO 230
      PERCNT=DIFF/TOTL2*100.
  230 RETURN
C     ..................................................................
C
C     ---PRINT RESULTS---
C     ************************
      ENTRY CWRITE
C     ************************
C
      WRITE (6,260) STOR,QREFLX,STORT,CFLUX,QRET,PUMP,CFLUXT,ETFLUX,CHST
     1,FLXPT,CHD2,TOTL1,CHD1,FLUX,FLUXS,ETFLXT,CHDT,SUMR,PUMPT,FLXNT,TOT
     2L2,DIFF,PERCNT
      IF (NCH.EQ.0) GO TO 240
      WRITE (6,270)
      WRITE (6,280) ((JFLO(I,J),J=1,3),FLOW(I),I=1,NCH)
C
C     ---COMPUTE VERTICAL FLOW---
  240 X=0.
      Y=0.
      IF (K0.EQ.1) RETURN
      DO 250 I=2,I1
      DO 250 J=2,J1
      X=X+(PHI(I,J,1)-PHI(I,J,2))*TK(I,J,1)*DELX(J)*DELY(I)*2./(DELZ(1)+
     1DELZ(2))
  250 Y=Y+(PHI(I,J,K1)-PHI(I,J,K0))*TK(I,J,K1)*DELX(J)*DELY(I)*2./(DELZ(
     1K1)+DELZ(K0))
      WRITE (6,290) Y,X
      RETURN
C
C     ---FORMATS---
C
C     ------------------------------------------------------------------
C
C
C
  260 FORMAT ('0',10X,'CUMULATIVE MASS BALANCE:',16X,'L**3',23X,'RATES F
     1OR THIS TIME STEP:',16X,'L**3/T'/11X,24('-'),43X,25('-')//20X,'SOU
     2RCES:',69X,'STORAGE =',F20.4/20X,8('-'),68X,'RECHARGE =',F20.4/27X
     3,'STORAGE =',F20.2,35X,'CONSTANT FLUX =',F20.4/26X,'RECHARGE =',F2
     40.2,41X,'PUMPING =',F20.4/21X,'CONSTANT FLUX =',F20.2,30X,'EVAPOTR
     5ANSPIRATION =',F20.4/21X,'CONSTANT HEAD =',F20.2,34X,'CONSTANT HEA
```

```
     6D:'/27X,'LEAKAGE =',F20.2,46X,'IN =',F20.4/21X,'TOTAL SOURCES =',F
     720.2,45X,'OUT =',F20.4/96X,'LEAKAGE:'/20X,'DISCHARGES:',45X,'FROM 
     8PREVIOUS PUMPING PERIOD =',F20.4/20X,11('-'),68X,'TOTAL =',F20.4/1
     96X,'EVAPOTRANSPIRATION =',F20.2/21X,'CONSTANT HEAD =',F20.2,36X,'S
     SUM OF RATES =',F20.4/19X'QUANTITY PUMPED =',F20.2/27X,'LEAKAGE =',
     $F20.2/19X,'TOTAL DISCHARGE =',F20.2//17X,'DISCHARGE-SOURCES =',F20
     $.2/15X,'PER CENT DIFFERENCE =',F20.2//)
 270 FORMAT ('0FLOW RATES TO CONSTANT HEAD NODES:'/' ',34('-')//' ',3(9
     1X,'K',4X,'I',4X,'J',5X,'RATE (L**3/T)')/' ',3(9X,'-',4X,'-',4X,'-'
     2,5X,13('-'))/)
 280 FORMAT (/(1X,3(I10,2I5,G18.7)))
 290 FORMAT ('0FLOW TO TOP LAYER =',G15.7,'   FLOW TO BOTTOM LAYER =',G
     115.7,'       POSITIVE UPWARD')
      END
```

```
      SUBROUTINE PRNTAI(PHI,STRT,T,S,WELL,DELX,DELY)
C     ------------------------------------------------------------------
C     PRINT MAPS OF DRAWDOWN AND HYDRAULIC HEAD
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI,Z,XLABEL,YLABEL,TITLE,XN1,MESUR
      REAL *4K
C
      DIMENSION PHI(I0,J0,K0), STRT(I0,J0,K0), S(I0,J0,K0), WELL(I0,J0,K
     10), DELX(J0), DELY(I0), T(I0,J0,K0)
C
      COMMON /INTEGR/ I0,J0,K0,I1,J1,K1,I,J,K,NPER,KTH,ITMAX,LENGTH,KP,N
     1WEL,NUMT,IFINAL,IT,KT,IHEAD,IDRAW,IFLO,IERR,I2,J2,K2,IMAX,ITMX1,NC
     2H,IDK1,IDK2,IWATER,IQRE,IP,JP,IQ,JQ,IK,JK,K5,IPU1,IPU2,ITK
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANK
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2
      RETURN
C     ..................................................................
C
C     ---INITIALIZE VARIABLES FOR PLOT---
C     ********************
      ENTRY MAP
C     ********************
      YDIM=0.
      WIDTH=0.
      DO 10 J=2,J1
   10 WIDTH=WIDTH+DELX(J)
      DO 20 I=2,I1
   20 YDIM=YDIM+DELY(I)
   30 XSF=DINCH*XSCALE
      YSF=DINCH*YSCALE
      NYD=YDIM/YSF
      IF (NYD*YSF.LE.YDIM-DELY(I1)/2.) NYD=NYD+1
      IF (NYD.LE.12) GO TO 40
      DINCH=YDIM/(12.*YSCALE)
      WRITE (6,330) DINCH
      IF (YSCALE.LT.1.0) WRITE (6,340)
      GO TO 30
   40 NXD=WIDTH/XSF
      IF (NXD*XSF.LE.WIDTH-DELX(J1)/2.) NXD=NXD+1
      N4=NXD*N1+1
      N5=NXD+1
      N6=NYD+1
      N8=N2*NYD+1
      NA(1)=N4/2-1
      NA(2)=N4/2
      NA(3)=N4/2+3
      NC=(N3-N8-10)/2
      ND=NC+N8
      NE=MAX0(N5,N6)
      VF1(3)=DIGIT(ND)
      VF2(3)=DIGIT(ND)
      VF3(3)=DIGIT(NC)
      XLABEL(3)=MESUR
```

```
      YLABEL(6)=MESUR                                                   PRN 570
      DO 60 I=1,NE                                                      PRN 580
      NNX=N5-I                                                          PRN 590
      NNY=I-1                                                           PRN 600
      IF (NNY.GE.N6) GO TO 50                                           PRN 610
      YN(I)=YSF*NNY/YSCALE                                              PRN 620
   50 IF (NNX.LT.0) GO TO 60                                            PRN 630
      XN(I)=XSF*NNX/YSCALE                                              PRN 640
   60 CONTINUE                                                          PRN 650
      RETURN                                                            PRN 660
C     ..................................................................PRN 670
C                                                                       PRN 680
C     ********************                                              PRN 690
      ENTRY PRNTA(NG,LA)                                                PRN 700
C     ********************                                              PRN 710
C     ---VARIABLES INITIALIZED EACH TIME A PLOT IS REQUESTED---         PRN 720
      DIST=WIDTH-DELX(J1)/2.                                            PRN 730
      JJ=J1                                                             PRN 740
      LL=1                                                              PRN 750
      Z=NXD*XSF                                                         PRN 760
      IF (NG.EQ.1) WRITE (6,300) (TITLE(I),I=1,3),LA                    PRN 770
      IF (NG.EQ.2) WRITE (6,300) (TITLE(I),I=4,6),LA                    PRN 780
      DO 290 I=1,N4                                                     PRN 790
C                                                                       PRN 800
C     ---LOCATE X AXES---                                               PRN 810
      IF (I.EQ.1.OR.I.EQ.N4) GO TO 70                                   PRN 820
      PRNT(1)=SYM(12)                                                   PRN 830
      PRNT(N8)=SYM(12)                                                  PRN 840
      IF ((I-1)/N1*N1.NE.I-1) GO TO 90                                  PRN 850
      PRNT(1)=SYM(14)                                                   PRN 860
      PRNT(N8)=SYM(14)                                                  PRN 870
      GO TO 90                                                          PRN 880
C                                                                       PRN 890
C     ---LOCATE Y AXES---                                               PRN 900
   70 DO 80 J=1,N8                                                      PRN 910
      IF ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)                           PRN 920
   80 IF ((J-1)/N2*N2.NE.J-1) PRNT(J)=SYM(13)                           PRN 930
C                                                                       PRN 940
C     ---COMPUTE LOCATION OF NODES AND DETERMINE APPROPRIATE SYMBOL---  PRN 950
   90 IF (DIST.LT.0..OR.DIST.LT.Z-XN1*XSF) GO TO 240                    PRN 960
      YLEN=DELY(2)/2.                                                   PRN 970
      DO 220 L=2,I1                                                     PRN 980
      J=YLEN*N2/YSF+1.5                                                 PRN 990
      IF (T(L,JJ,LA).EQ.0.) GO TO 160                                   PRN1000
      IF (S(L,JJ,LA).LT.0.) GO TO 210                                   PRN1010
      INDX3=0                                                           PRN1020
      GO TO (100,110), NG                                               PRN1030
  100 K=(STRT(L,JJ,LA)-PHI(L,JJ,LA))*FACT1                              PRN1040
C     -TO CYCLE SYMBOLS FOR DRAWDOWN, REMOVE C FROM COL. 1 OF NEXT CARD-PRN1050
C     K=AMOD(K,10.)                                                     PRN1060
      GO TO 120                                                         PRN1070
  110 K=PHI(L,JJ,LA)*FACT2                                              PRN1080
  120 IF (K) 130,160,140                                                PRN1090
  130 IF (J-2.GT.0) PRNT(J-2)=SYM(13)                                   PRN1100
      N=-K+.5                                                           PRN1110
      IF (N.LT.100) GO TO 150                                           PRN1120
```

```
      GO TO 190                                                         PRN1130
  140 N=K+.5                                                            PRN1140
      IF (N.LT.100) GO TO 150                                           PRN1150
      IF (N.GT.999) GO TO 190                                           PRN1160
      INDX3=N/100                                                       PRN1170
      IF (J-2.GT.0) PRNT(J-2)=SYM(INDX3)                                PRN1180
      N=N-INDX3*100                                                     PRN1190
  150 INDX1=MOD(N,10)                                                   PRN1200
      IF (INDX1.EQ.0) INDX1=10                                          PRN1210
C     -TO CYCLE SYMBOLS FOR DRAWDOWN, REMOVE C FROM COL. 1 OF NEXT CARD-PRN1220
C     IF (NG.EQ.1) GO TO 170                                            PRN1230
      INDX2=N/10                                                        PRN1240
      IF (INDX2.GT.0) GO TO 180                                         PRN1250
      INDX2=10                                                          PRN1260
      IF (INDX3.EQ.0) INDX2=15                                          PRN1270
      GO TO 180                                                         PRN1280
  160 INDX1=15                                                          PRN1290
  170 INDX2=15                                                          PRN1300
  180 IF (J-1.GT.0) PRNT(J-1)=SYM(INDX2)                                PRN1310
      PRNT(J)=SYM(INDX1)                                                PRN1320
      GO TO 220                                                         PRN1330
  190 DO 200 II=1,3                                                     PRN1340
      JI=J-3+II                                                         PRN1350
  200 IF (JI.GT.0) PRNT(JI)=SYM(11)                                     PRN1360
  210 IF (S(L,JJ,LA).LT.0.) PRNT(J)=SYM(16)                             PRN1370
  220 YLEN=YLEN+(DELY(L)+DELY(L+1))/2.                                  PRN1380
  230 DIST=DIST-(DELX(JJ)+DELX(JJ-1))/2.                                PRN1390
      JJ=JJ-1                                                           PRN1400
      IF (JJ.EQ.0) GO TO 240                                            PRN1410
      IF (DIST.GT.Z-XN1*XSF) GO TO 230                                  PRN1420
  240 CONTINUE                                                          PRN1430
C                                                                       PRN1440
C     ---PRINT AXES,LABELS, AND SYMBOLS---                              PRN1450
      IF (I-NA(LL).EQ.0) GO TO 260                                      PRN1460
      IF ((I-1)/N1*N1-(I-1)) 270,250,270                                PRN1470
  250 WRITE (6,VF1) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XN(1+(I-1)/6)    PRN1480
      GO TO 280                                                         PRN1490
  260 WRITE (6,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XLABEL(LL)       PRN1500
      LL=LL+1                                                           PRN1510
      GO TO 280                                                         PRN1520
  270 WRITE (6,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8)                  PRN1530
C                                                                       PRN1540
C     ---COMPUTE NEW VALUE FOR Z AND INITIALIZE PRNT---                 PRN1550
  280 Z=Z-2.*XN1*XSF                                                    PRN1560
      DO 290 J=1,N8                                                     PRN1570
  290 PRNT(J)=SYM(15)                                                   PRN1580
C                                                                       PRN1590
C     ---NUMBER AND LABEL Y AXIS AND PRINT LEGEND---                    PRN1600
      WRITE (6,VF3) (BLANK(J),J=1,NC),(YN(I),I=1,N6)                    PRN1610
      WRITE (6,320) (YLABEL(I),I=1,6)                                   PRN1620
      IF (NG.EQ.1) WRITE (6,310) FACT1                                  PRN1630
      IF (NG.EQ.2) WRITE (6,310) FACT2                                  PRN1640
      RETURN                                                            PRN1650
C                                                                       PRN1660
C     ---FORMATS---                                                     PRN1670
C                                                                       PRN1680
```

```
C     ------------------------------------------------------------------PRN1690
C                                                                       PRN1700
C                                                                       PRN1710
  300 FORMAT ('1',49X,3A8,'LAYER',I4//)                                 PRN1720
  310 FORMAT ('0EXPLANATION'/' ',11('-')//' R = CONSTANT HEAD BOUNDARY'/PRN1730
     1' *** = VALUE EXCEEDED 3 FIGURES'/' MULTIPLICATION FACTOR =',F8.3)PRN1740
  320 FORMAT ('0',39X,6A8)                                              PRN1750
  330 FORMAT ('0',25X,10('*'),' TO FIT MAP WITHIN 12 INCHES, DINCH REVISPRN1760
     1ED TO',G15.7,1X,10('*'))                                          PRN1770
  340 FORMAT ('0',45X,'NOTE: GENERALLY SCALE SHOULD BE > OR = 1.0')     PRN1780
      END                                                               PRN1790
```

```
      BLOCK DATA
C     ----------
C
C     SPECIFICATIONS:
      REAL *8XLABEL,YLABEL,TITLE,XN1,MESUR
C
      COMMON /SARRAY/ ICHK(13),LEVEL1(9),LEVEL2(9)
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(6),XN1,MESUR,PRNT(122),BLANK
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2
C     *****************************************************************
C
      DATA ICHK/'DRAW','HEAD','MASS','DK1','DK2','WATE','RECH','PUN1','P
     1UN2','ITKR',3*0/
      DATA SYM/'1','2','3','4','5','6','7','8','9','0','*','|','-','+','
     1 ','R','W'/
      DATA PRNT/122*' '/,N1,N2,N3,XN1/6,10,133,.833333333D-1/,BLANK/60*'
     1 '/,NA(4)/1000/
      DATA XLABEL/' X DIS- ','TANCE IN','  MILES '/,YLABEL/'DISTANCE',' 
     1FROM OR','IGIN IN ','Y DIRECT','ION, IN ','MILES   '/,TITLE/'PLOT 
     2OF ','DRAWDOWN',' ','PLOT OF ','HYDRAULI','C HEAD'/
      DATA DIGIT/'1','2','3','4','5','6','7','8','9','10','11','12','13'
     1,'14','15','16','17','18','19','20','21','22','23','24','25','26',
     2'27','28','29','30','31','32','33','34','35','36','37','38','39','
     340','41','42','43','44','45','46','47','48','49','50','51','52','5
     43','54','55','56','57','58','59','60','61','62','63','64','65','66
     5','67','68','69','70','71','72','73','74','75','76','77','78',' 79
     6','80','81','82','83','84','85','86','87','88','89','90','91','92'
     7,'93','94','95','96','97','98','99','100','101','102','103',' 104'
     8,'105','106','107','108','109','110','111','112','113','114','115'
     9,'116','117','118','119','120','121','122'/
      DATA VF1/'(1H ',',',     ','A1,F','10.2',')'/
      DATA VF2/'(1H ',',',     ','A1,1','X,A8',')'/
      DATA VF3/'(1H0',',',     ','A1,F','3.1,','12F1','0.2)'/
C     *****************************************************************
      END
```