USGS
science for a changing world

U.S. DEPARTMENT OF THE INTERIOR
MARCH 3, 1849

Techniques of Water-Resources Investigations
of the United States Geological Survey

Chapter C1

FINITE-DIFFERENCE MODEL FOR AQUIFER SIMULATION IN TWO DIMENSIONS WITH RESULTS OF NUMERICAL EXPERIMENTS

By P. C. Trescott, G. F. Pinder, and S. P. Larson

Book 7

AUTOMATED DATA PROCESSING AND COMPUTATIONS

Click here to return to USGS Publications

UNITED STATES DEPARTMENT OF THE INTERIOR

CECIL D. ANDRUS, Secretary

GEOLOGICAL SURVEY

H. William Menard, Director

First printing 1976

Second printing 1980

Requests, at cost, for the Card Deck listed in Attachment VII should be directed to:
Ralph N. Eicher, Chief, Office of Teleprocessing, M.S. 805, National Center,
U.S. Geological Survey, Reston, Virginia 22092.

UNITED STATES GOVERNMENT PRINTING OFFICE, WASHINGTON : 1976

For sale by the Branch of Distribution, U.S. Geological Survey,
1200 South Eads Street, Arlington, VA 22202

PREFACE

The series of manuals on techniques describes procedures for planning and executing specialized work in water-resources investigations. The material is grouped under major headings called books and further subdivided into sections and chapters; section C of Book 7 is on computer programs.

"Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments" supersedes the report published in 1970 entitled, "A digital model for aquifer evaluation" by G. F. Pinder as Chapter C1 of Book 7. The new Chapter C1 represents a significant improvement in the computational capability to solve the flow equations and has greater flexibility in the hydrologic situations that can be simulated.

CONTENTS

FIGURES

TABLES

FINITE-DIFFERENCE MODEL FOR AQUIFER SIMULATION IN TWO DIMENSIONS WITH RESULTS OF NUMERICAL EXPERIMENTS

By P. C. Trescott, G. F. Pinder, and S. P. Larson

Abstract

The model will simulate ground-water flow in an artesian aquifer, a water-table aquifer, or a combined artesian and water-table aquifer. The aquifer may be heterogeneous and anisotropic and have irregular boundaries. The source term in the flow equation may include well discharge, constant recharge, leakage from confining beds in which the effects of storage are considered, and evapotranspiration as a linear function of depth to water.

The theoretical development includes presentation of the appropriate flow equations and derivation of the finite-difference approximations (written for a variable grid). The documentation emphasizes the numerical techniques that can be used for solving the simultaneous equations and describes the results of numerical experiments using these techniques. Of the three numerical techniques available in the model, the strongly implicit procedure, in general, requires less computer time and has fewer numerical difficulties than do the iterative alternating direction implicit procedure and line successive overrelaxation (which includes a two-dimensional correction procedure to accelerate convergence).

The documentation includes a flow chart, program listing, an example simulation, and sections on designing an aquifer model and requirements for data input. It illustrates how model results can be presented on the line printer and pen plotters with a program that utilizes the graphical display software available from the Geological Survey Computer Center Division. In addition the model includes options for reading input data from a disk and writing intermediate results on a disk.

Introduction

The finite-difference aquifer model documented in this report is designed to simulate in two dimensions the response of an aquifer to an imposed stress. The aquifer may be artesian, water table, or a combination of artesian and water table; it may be heterogeneous and anisotropic and have irregular boundaries. The model permits leakage from confining beds in which the effects of storage are considered, constant recharge, evapotranspiration as a linear function of depth to water, and well discharge. Although it was not designed for cross-sectional problems, the model has been used with some success for this type of simulation.

The aquifer simulator has evolved from Pinder's (1970) original model and modifications by Pinder (1969) and Trescott (1973). The model documented by Trescott (1973) incorporates several features described by Prickett and Lonnquist (1971) and has been applied to a variety of aquifer simulation problems by various users. The model described in this report is basically the same as the 1973 version but includes minor modifications to the logic and data input. In addition, the user may choose an equation solving scheme from among the alternating direction implicit procedure, line successive overrelaxation, and the strongly implicit procedure. The program is arranged so that other techniques for solving simultaneous equations can be coded and substituted for the iterative techniques included with the model.

The documentation is intended to be reasonably self contained, but it assumes that the user has an elementary knowledge of the physics of ground-water flow, finite-difference methods of solving partial differential equations, matrix algebra, and the FORTRAN IV language.

Theoretical Development

Ground-water flow equation

The partial differential equation of ground-water flow in a confined aquifer in two dimensions may be written as

$$\frac{\partial}{\partial x}(T_{xx}\frac{\partial h}{\partial x})+\frac{\partial}{\partial x}(T_{xy}\frac{\partial h}{\partial y})+\frac{\partial}{\partial y}(T_{yx}\frac{\partial h}{\partial x})+\frac{\partial}{\partial y}(T_{yy}\frac{\partial h}{\partial y})=S\frac{\partial h}{\partial t}+W(x,y,t) \qquad (1)$$

in which

T_{xx}, T_{xy}, T_{yx}, T_{yy} are the components of the transmissivity tensor (L^2t^{-1});

h is hydraulic head (L);

S is the storage coefficient (dimensionless);

$W(x, y, t)$ is the volumetric flux of recharge or withdrawal per unit surface area of the aquifer (Lt^{-1}).

The reader is referred to Pinder and Bredehoeft (1968) for development and discussion of equation 1. In the simulation model, equation 1 is simplified by assuming that the Cartesian coordinate axes x and y are alined with the principal components of the transmissivity tensor, T_{xx} and T_{yy}, giving

$$\frac{\partial}{\partial x}(T_{xx}\frac{\partial h}{\partial x})+\frac{\partial}{\partial y}(T_{yy}\frac{\partial h}{\partial y})=S\frac{\partial h}{\partial t}+W(x,y,t). \qquad (2)$$

In water-table aquifers, transmissivity is a function of head. Assuming that the coordinate axes are co-linear with the principal components of the hydraulic conductivity tensor, the flow equation may be expressed as (Bredehoeft and Pinder, 1970)

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

in which

K_{xx}, K_{yy} are the principal components of the hydraulic conductivity tensor (Lt^{-1});

S_y is the specific yield of the aquifer (dimensionless);

b is the saturated thickness of the aquifer (L).

Finite-difference approximations

In order to solve equation 2 or 3 for a heterogeneous aquifer with irregular boundaries, one approach is to subdivide the region into rectangular blocks in which the aquifer properties are assumed to be uniform. The continuous derivatives in equations 2 and 3 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 aquifer.

Utilizing a block-centered, finite-difference grid in which variable grid spacing is permitted (fig. 1), equation 2 may be approximated as

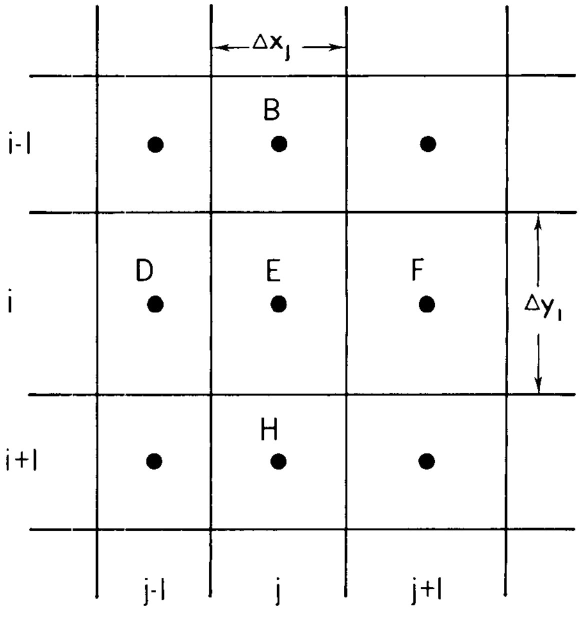


FIGURE 1.—Index scheme for finite-difference grid and coefficients of finite-difference equation written for node (*i, j*).

$$\frac{1}{\Delta x_j}\left[\left(T_{xx}\frac{\partial h}{\partial x}\right)_{i,j+\frac{1}{2}}-\left(T_{xx}\frac{\partial h}{\partial x}\right)_{i,j-\frac{1}{2}}\right]+\frac{1}{\Delta y_i}\left[\left(T_{yy}\frac{\partial h}{\partial y}\right)_{i+\frac{1}{2},j}-\left(T_{yy}\frac{\partial h}{\partial y}\right)_{i-\frac{1}{2},j}\right]=\frac{S_{i,j}}{\Delta t}\left(h_{i,j,k}-h_{i,j,k-1}\right)+W_{i,j,k} \qquad (4)$$

in which

Δx_j is the space increment in the x direction for column j as shown in figure 1 (L);

Δy_i is in the space increment in the y direction for row i as shown in figure 1 (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 time index.

Equation 4 may be approximated again as

$$\frac{1}{\Delta x_j}\left\{\left[T_{xx\,(i,j+\frac{1}{2})}\frac{(h_{i,j+1,k}-h_{i,j,k})}{\Delta x_{j+\frac{1}{2}}}\right]-\left[T_{xx\,(i,j-\frac{1}{2})}\frac{(h_{i,j,k}-h_{i,j-1,k})}{\Delta x_{j-\frac{1}{2}}}\right]\right\}+\frac{1}{\Delta y_i}\left\{\left[T_{yy\,(i+\frac{1}{2},j)}\frac{(h_{i+1,j,k}-h_{i,j,k})}{\Delta y_{i+\frac{1}{2}}}\right]-\left[T_{yy\,(i-\frac{1}{2},j)}\frac{(h_{i,j,k}-h_{i-1,j,k})}{\Delta y_{i-\frac{1}{2}}}\right]\right\}=\frac{S_{i,j}}{\Delta t}\left(h_{i,j,k}-h_{i,j,k-1}\right)+W_{i,j,k} \qquad (5)$$

in which

$T_{xx(i,j+\frac{1}{2})}$ is the transmissivity between node (i,j) and node $(i,j+1)$;

$\Delta x_{j+\frac{1}{2}}$ is the distance between node (i,j) and node $(i,j+1)$.

Equation 5 is written implicitly, that is, the head values on the left-hand side are at the new (k) time level. Following a convention similar to that introduced by Stone (1968), the notation in equation 5 may be simplified by writing

$$F_{i,j}(h_{i,j+1,k}-h_{i,j,k})-D_{i,j}(h_{i,j,k}-h_{i,j-1,k})+H_{i,j}(h_{i+1,j,k}-h_{i,j,k})-B_{i,j}(h_{i,j,k}-h_{i-1,j,k})=\frac{S_{i,j}}{\Delta t}(h_{i,j,k}-h_{i,j,k-1})+W_{i,j,k} \qquad (6)$$

in which

$$B_{i,j}=\frac{\left[\dfrac{2T_{yy\,[i,j]}\,T_{yy\,[i-1,j]}}{T_{yy\,[i,j]}\,\Delta y_{i-1}+T_{yy\,[i-1,j]}\,\Delta y_i}\right]}{\Delta y_i}. \qquad (7a)$$

The term in brackets is the harmonic mean of

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

It represents the ratio $T_{yy(i-\frac{1}{2})}/\Delta y_{i-\frac{1}{2}}$ in equation 5.

Similarly,

$$D_{i,j}=\frac{\left[\dfrac{2T_{xx[i,j]}\,T_{xx\,[i,j-1]}}{T_{xx\,[i,j]}\,\Delta x_{j-1}+T_{xx[i,j-1]}\,\Delta x_j}\right]}{\Delta x_j}; \qquad (7b)$$

$$F_{i,j}=\frac{\left[\dfrac{2T_{xx[i,j]}\,T_{xx\,[i,j+1]}}{T_{xx[i,j]}\Delta x_{j+1}+T_{xx\,[i,j+1]}\Delta x_j}\right]}{\Delta x_j}; \qquad (7c)$$

$$H_{i,j}=\frac{\left[\dfrac{2T_{yy[i+1,j]}\,T_{yy\,[i,j]}}{T_{yy[i,j]}\,\Delta y_{i+1}+T_{yy[i+1,j]}\Delta y_i}\right]}{\Delta y_i}. \qquad (7d)$$

Use of the harmonic mean (1) insures continuity across cell boundaries at steady state if a variable grid is used, and (2) makes the appropriate coefficients zero at no-flow boundaries.

Equation 6 is also used to approximate equation 3 by replacing S with Sy and defining the transmissivities in equations 7a through 7d as a function of the head from the preceding iteration. As an example,

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

in which n is the iteration index.

The notation may be simplified further by omitting subscripts not including a "+1" or "−1" (except where necessary for clarity) and by following the convention that unknown terms are placed on the left-hand side of the equations. Equation 6 may be rearranged and expressed as

$$Bh_{i-1}+Dh_{j-1}+Eh+Fh_{j+1}+Hh_{i+1}=Q \quad (8)$$

in which

$$E=-\left(B+D+F+H+\frac{S}{\Delta t}\right);$$

$$Q=-\frac{S}{\Delta t}h_{k-1}+W.$$

Source term

The source term $W(x,y,t)$ can include well discharge, transient leakage from a confining bed, recharge from precipitation and evapotranspiration. In the model the source term is computed as

$$W_{j,j,k}=\frac{Q_{w\,[i,j,k]}}{\Delta x_j\ \Delta y_i}-q_{re\,[i,j,k]}-q'_{i,j,k}+q_{et\,[i,j,k]}$$

in which

$Q_{w\,[i,j,k]}$ is the well discharge (L^3t^{-1});

$q_{rc[i,j,k]}$ is the recharge flux per unit area (Lt^{-1});

$q'_{i,j,k}$ is the flux per unit area from a confining layer (Lt^{-1});

$q_{et\,[i,j,k]}$ is the evapotranspiration flux per unit area (Lt^{-1}).

Leakage

Leakage from a confining layer or streambed in which storage is considered may be approximated by

$$q'_{i,j,k}\cong(h_{i,j,0}-h_{i,j,k})\frac{K'_{i,j}}{\left(\dfrac{\pi K'_{i,j}t}{3m^2_{i,j}S_{s[i,j]}}\right)^{1/2}m_{i,j}}\cdot\left\{1+2\sum_{n=1}^{\infty}\exp\left[\frac{-n^2}{\left(\dfrac{K'_{i,j}t}{3m^2_{i,j}S_{s[i,j]}}\right)}\right]\right\}+\frac{K'_{i,j}}{m_{i,j}}(\hat{h}_{i,j,0}-h_{i,j,0}) \quad (9)$$

in which

$h_{i,j,0}$ is the hydraulic head in the aquifer at the start of the pumping period (L);

$\hat{h}_{i,j,0}$ is the hydraulic head on the other side of the confining bed (L);

$K'_{i,j}$ is the hydraulic conductivity of the confining bed (L/t);

$m_{i,j}$ is the thickness of the confining bed (L);

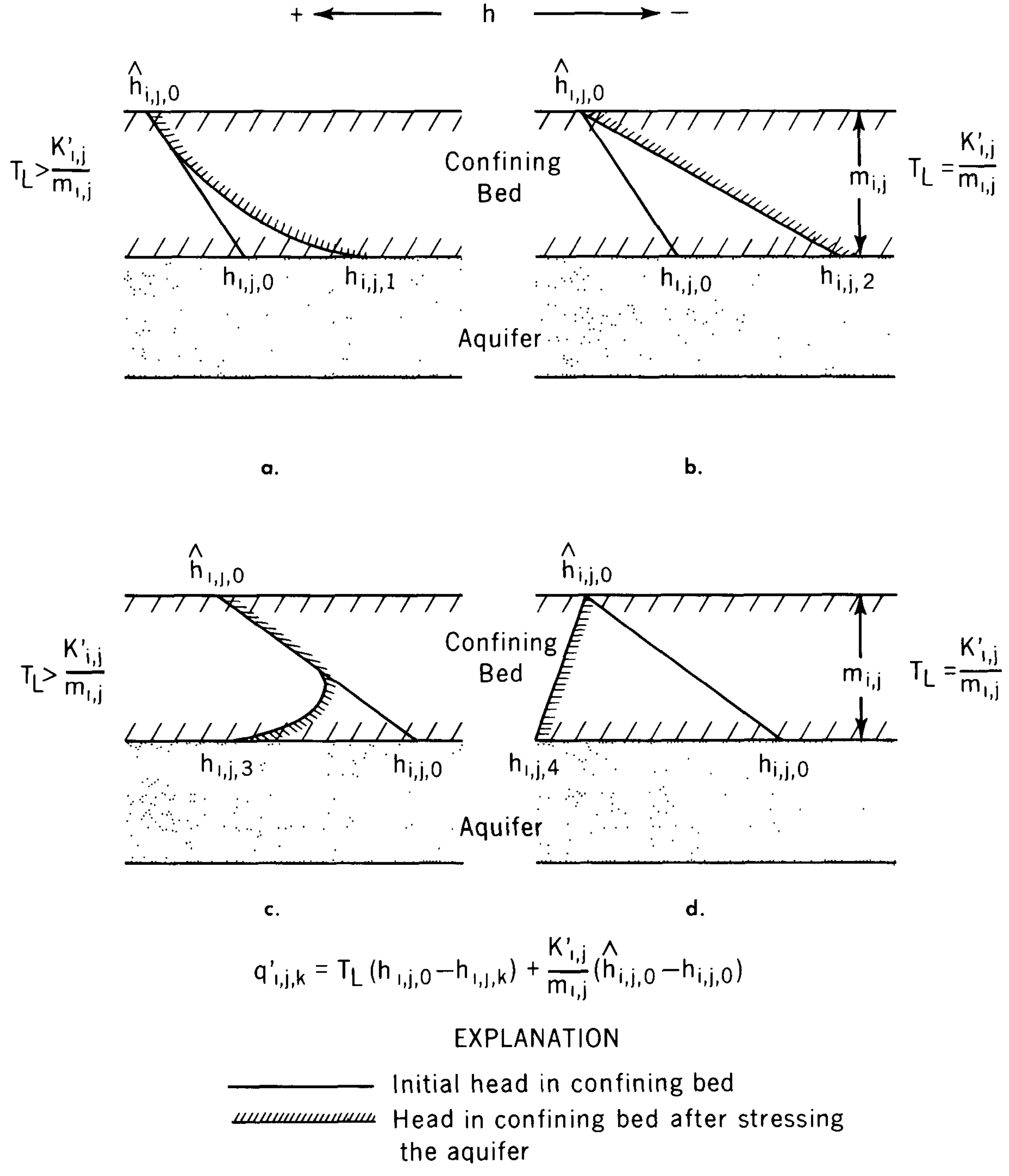
$S_{s[i,j]}$ is the specific storage in the confining layer (L^{-1});

$(K'_{i,j}t/m^2_{i,j}S_{s[i,j]})$ is dimensionless time; see Bredehoeft and Pinder (1970) for a discussion of leakage versus dimensionless time;

t is the elapsed time of the pumping period (t).

Equation 9 is modified from Bredehoeft and Pinder (1970, p. 887); note that it is the sum of two terms; the first term on the right-hand side of equation 9 considers transient effects; the second term is steady leakage due to the initial gradient across the confining bed. (See fig. 2.) Figure 2 illustrates the head distribution in the confining layer at any given point in the aquifer system at two different times in each of two successive pumping periods. (The succession of head values in the aquifer is shown by $h_{i,j,1}, \ldots h_{i,j,4}$.) The solid line represents the head distribution at the beginning of the pumping period; the gradient $((\hat{h}_{i,j,0}-h_{i,j,0})/m_{i,j})$ appears in the second term of equation 9. The hatchured line represents the head distribution in the confining bed after stressing the pumped aquifer and is a summation of the initial head distribution and the change in head distribution due to the stresses on the aquifer. The factor T_L in figure 2 represents the part of the first term in equation 9 independent of head (that is, the transient leakage coefficient).

In figure 2a the confining bed is assumed to have significant storage, pumping has lowered the head to $h_{i,j,1}$ and the net (or total) gradient is for some dimensionless time <0.5. After transient effects have dissipated, a uniform gradient across the confining bed is established. (See fig. 2b.) Then if the stress on the aquifer is changed by turning off pumping wells and starting recharge wells, the initial head distribution in the confining bed



$$q'_{i,j,k} = T_L\,(h_{i,j,0} - h_{i,j,k}) + \frac{K'_{i,j}}{m_{i,j}}(\hat{h}_{i,j,0} - h_{i,j,0})$$

FIGURE 2.—In the first pumping period, (a) illustrates the head distribution in the confining bed at one time when transient leakage effects are significant; (b) illustrates a time after transient effects have dissipated; in the second pumping period, (c) is analogous to (a) and (d) is analogous to (b).

for the new conditions is shown in figure 2c and is equal to the final distribution for the first pumping period. The net head distribution in figure 2c is affected by storage in the confining bed and is for some dimensionless time <0.5 (in the second pumping period). After storage effects have dissipated, the net gradient is shown in figure 2d.

For a simulation of several pumping periods, the program assumes that transient leakage effects from previous pumping periods have dissipated. This is accomplished at the start of each pumping period by initializing $h_{i,j,0}$ to the head at the end of the previous pumping period and setting t (and thereby dimensionless time) to zero (note that the parameter storing the cumulative simulation time is not affected). The assumption is reasonable if dimensionless time for previous pumping periods is at least 0.5 (Bredehoeft and Pinder, 1970, fig. 4) and can be checked by noting the value of dimensionless time printed in the output for the end of the previous pumping period. If the assumption is not valid, the code will need to be modified to include transient effects for one or more previous pumping periods.

In the model, equation 9 is used until dimensionless time reaches 3×10^{-3}; otherwise, the equation

$$q'_{i,j,k}\cong(h_{i,j,0}-h_{i,j,k})\frac{K'_{i,j}}{m_{i,j}}\left\{1+2\sum_{n=1}^{\infty}\exp\left[-n^2\pi^2\left(\frac{K'_{i,j}t}{3m^2_{i,j}S_{s[i,j]}}\right)\right]\right\}+\frac{K'_{i,j}}{m_{i,j}}(\hat{h}_{i,j,0}-h_{i,j,0}) \quad (10)$$

is used. Equation 10 is computationally more efficient for dimensionless times greater than about 3×10^{-3}.

The transient parts of equations 9 and 10 are based on the analytic solutions for the flux from a confining layer resulting from an instantaneous stepwise change in head in the aquifer. The factor of 1/3 appearing in dimensionless time is included in order to approximate the transient flux resulting from the actual drawdown in the aquifer. In effect the transient flux is approximated by applying a step change in head equal to the drawdown from the start of the pumping period at 1/3 of the elapsed time in the pumping period. (See fig. 3.)

The results of several numerical experiments indicate that it would be better to use

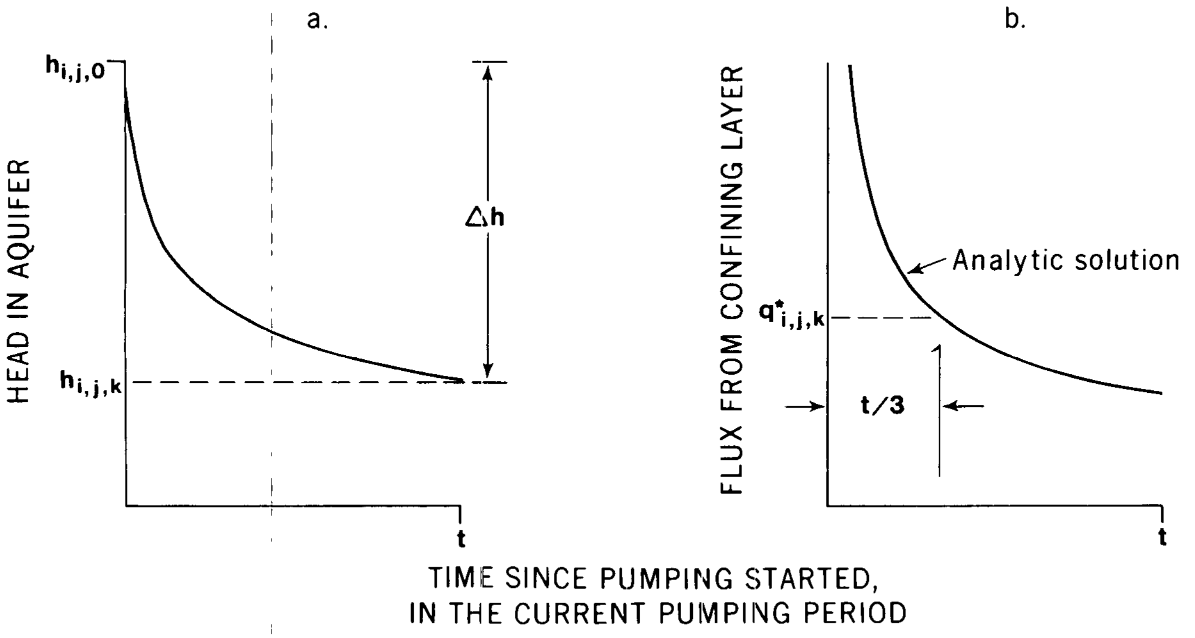


FIGURE 3.—The total drawdown at the elapsed time, t, in the pumping period (a) is applied at $t/3$ in equations 9 and 10 to approximate $q^*_{i,j,k}$, the transient part of $q'_{i,j,k}$ (b).

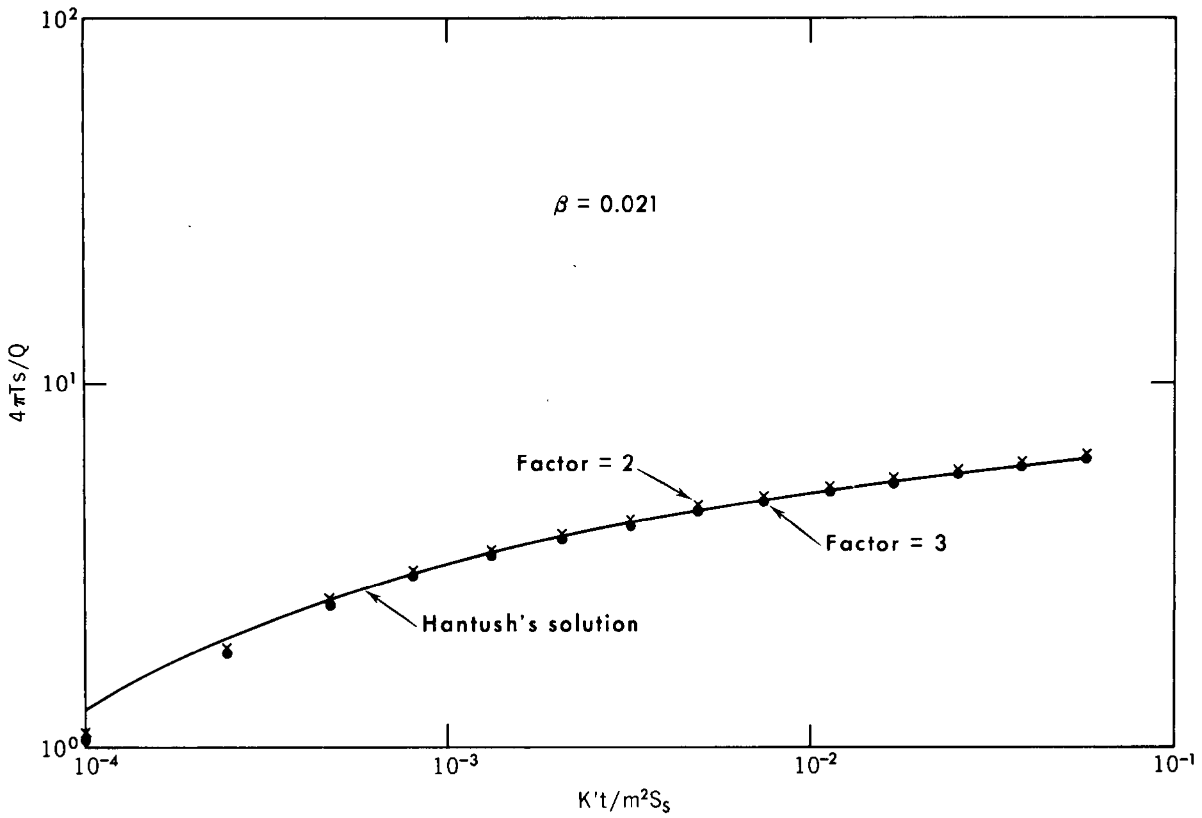


FIGURE 4.—Comparison of analytic solution and numerical results using factors of 2 and 3 in the transient leakage approximation.

a factor of 1/3 rather than the factor of 1/2 used in the approximation by Bredehoeft and Pinder (1970). In figure 4 are plotted numerical results and Hantush's (1960) analytic solution for $\beta = 0.021$ ($\beta = 0.25\ r\ [K'S_s/TS]^{1/2}$ and r is the radial distance from the center of the pumping well). The drawdown values using a factor of 1/3 are below but very close to the analytic curve after the first few time steps. The results using a factor of 1/2 are close to the analytic solution but are about twice as far above the analytic curve as the factor of 1/3 results are below the curve. In figure 5 are plotted the percent difference between the volume of leakage computed numerically and the volume determined analytically. Two sets of data are shown: a 14-step simulation between dimensionless times of 10^{-5} and 5.8×10^{-2} and an 11-step simulation between dimensionless times of 5.8×10^{-3} and 4.4×10^{-1}. Based on those experiments, if 4 or 5 time steps are simulated before the period of interest, the volume of leakage and the drawdown computed numerically using a factor 1/3 in equations 9 and 10 are close to the analytic solution.

Evapotranspiration

Evapotranspiration as a linear function of depth below the land surface is computed as

$$q_{et[i,j,k]} = \begin{cases} Q_{et} & [h_{i,j,k} \geqq G_{i,j}] \\ Q_{et} - \dfrac{Q_{et}}{ET_z}(G_{i,j} - h_{i,j,k}) & [ET_z > (G_{i,j} - h_{i,j,k})\ ;\ h_{i,j,k} < G_{i,j}] \\ 0 & [ET_z \leqq (G_{i,j} - h_{i,j,k})] \end{cases} \qquad (11)$$

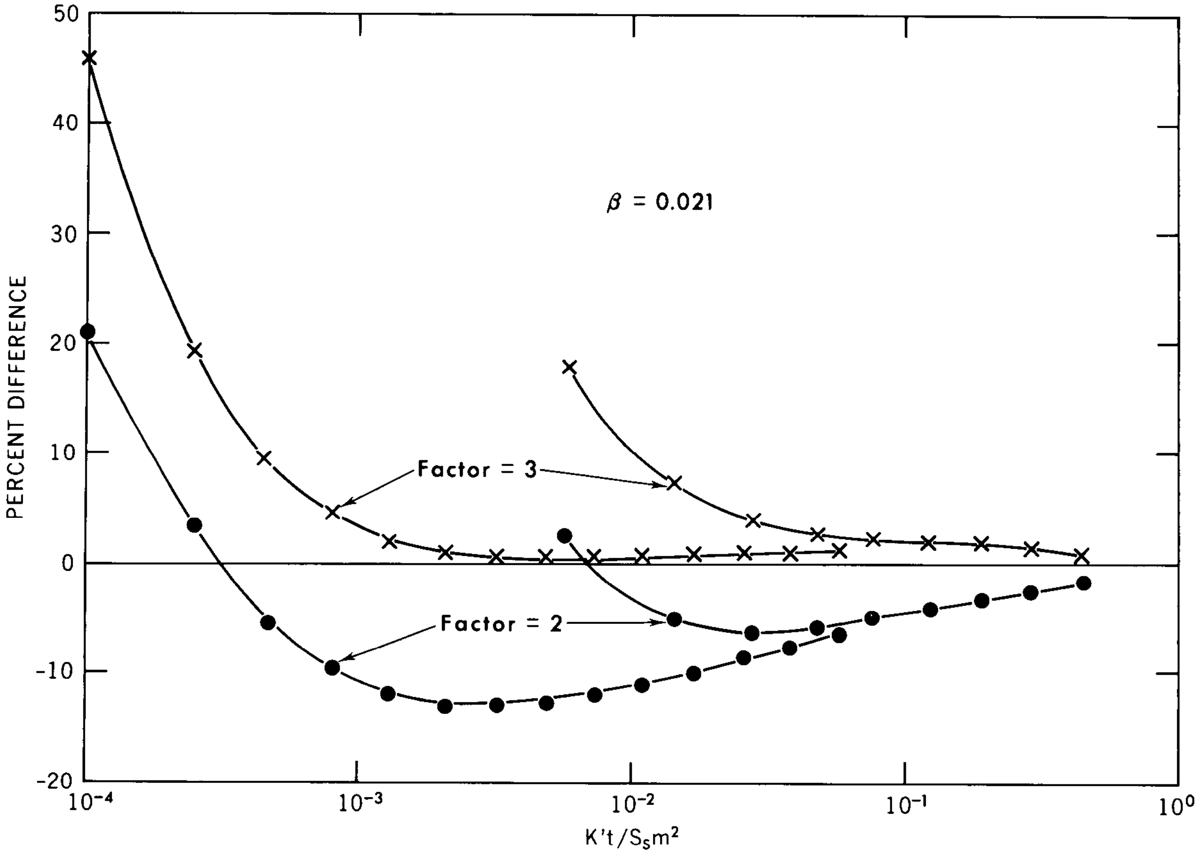


FIGURE 5.—Percent difference between the volume of leakage computed with the model approximation and Hantush's analytical results.

in which

Q_{et} is the maximum evapotranspiration rate (Lt^{-1});

ET_z is the depth below land surface at which evapotranspiration ceases (L);

$G_{i,j}$ is the elevation of the land surface (L).

This relationship (illustrated in fig. 6) is treated implicitly by separating the equation into two terms[1]: one term is included with the *E* coefficient on the left-hand side of equation 8; the other is a known term included in *Q* on the right-hand side of equation 8.

Other functions for evapotranspiration can be defined (for example, decreasing exponentially with depth), but it may be more difficult to treat these relationships numerically. The easiest approach is to make evapotranspiration an explicit function of the head at the previous iteration, but this may cause oscillations and difficulties with convergence. Normally, the oscillations may be dampened by making evapotranspiration a function of the head for the two previous iterations. A more sophisticated approach is to use the Newton-Raphson method, which is a rapidly converging iterative technique for treating systems of non-linear equations. (See, for example, Carnahan, Luther, and Wilkes, 1969, p. 319–329.)

Computation of head at the radius of a pumping well

The hydraulic head computed for a well node represents an average hydraulic head

---

[1] Some of the methods for implicit treatment of evapotranspiration, storage, and leakage have been adapted from Prickett and Lonnquist (1971).

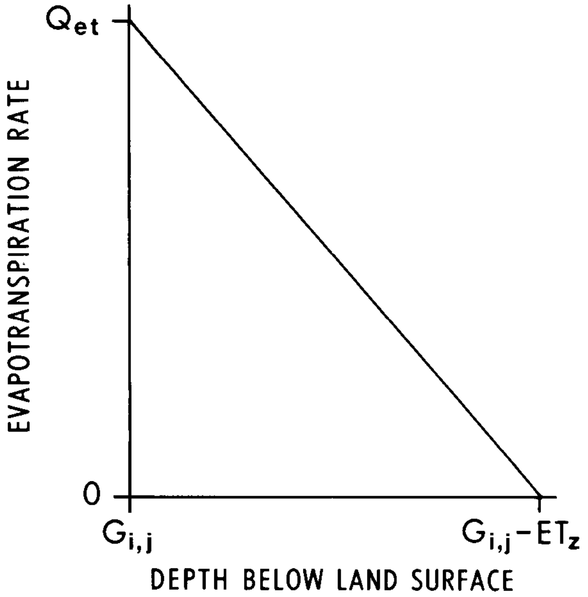


FIGURE 6.—Evapotranspiration decreases linearly from Q_{et} where the water table is at land surface to zero where the water table is less than or equal to $G_{i,j}-ET_z$.

computed for the block and is not the head in a well. An option to compute the head and drawdown at a well is included in the model. This computation uses the radius, r_e, of a hypothetical well for which the average value of head for the cell applies. An approximating equation is then used to make the extrapolation from r_e to the radius of a real well.

The radius r_e can be computed as (Prickett, 1967)

$$r_e = r_1/4.81 \tag{12}$$

in which $r_1 = \Delta x_j = \Delta y_i$ (fig. 7). Equation 12 assumes steady flow, no source term other than well discharge in the well block, and that the area around the well is isotropic and homogeneous. The derivation of equation 12 can be seen with reference to figure 7 in which the four nodes adjacent to node i,j are assumed to have head values equal to the value at node $i-1, j$. In figure 7a one-quarter of the discharge to the well node i,j is computed by the model as

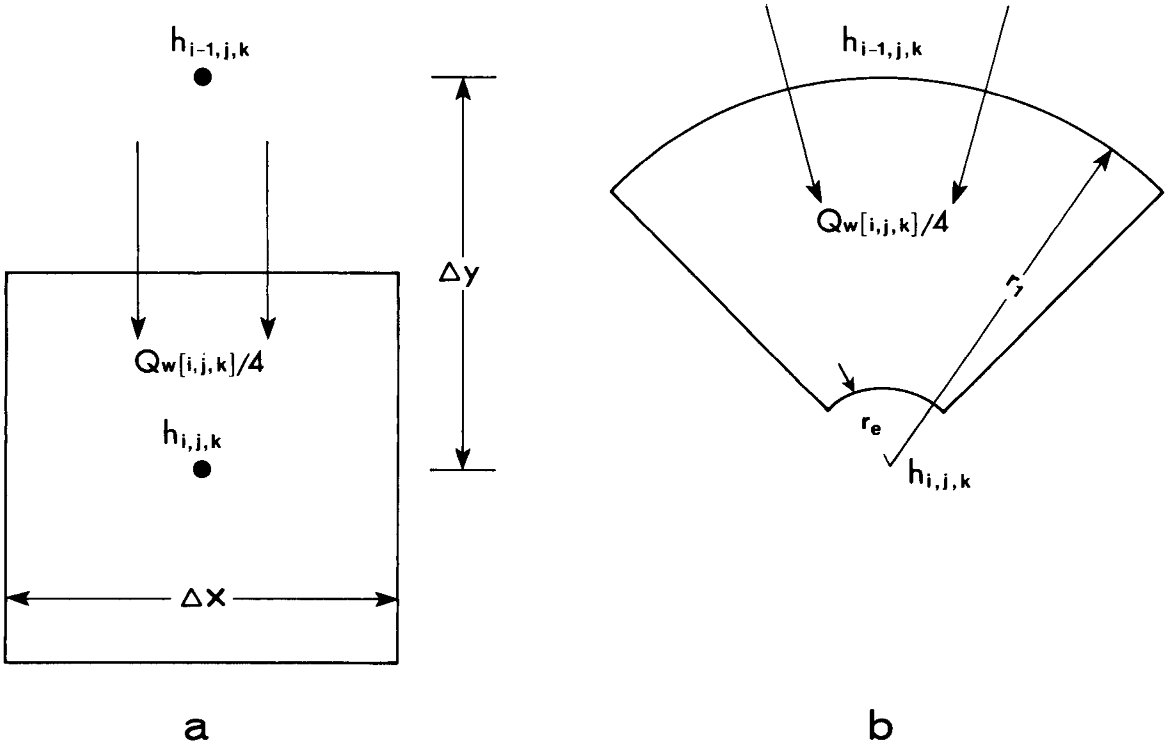


FIGURE 7.—Flow from cell $(i-1,j)$ to cell (i,j)(a) and equivalent radial flow to well (i,j) with radius r_e(b).

$$\frac{Q_{w[i,j,k]}}{4} = \Delta x_j T_{i,j} \frac{\Delta h}{\Delta y} \qquad (13)$$

in which

$\Delta h = h_{i-1,j,k} - h_{i,j,k}$;

$T_{i,j} = T_{xx[i,j]} = T_{yy[i,j]}$.

The equivalent discharge for radial flow to the well is given by the Thiem (1906) equation expressed as (see fig. 7b)

$$\frac{Q_{w[i,j,k]}}{4} = \frac{\pi T_{i,j}}{2} \frac{\Delta h}{\ln(r_1/r_e)}. \qquad (14)$$

Equating the discharges in equations 13 and 14 gives equation 12.

The Thiem equation is commonly used to extrapolate from the average hydraulic head for the cell at radius r_e to the head, h_w, at the desired well radius, r_w (Prickett and Lonnquist, 1971; Akbar, Arnold, and Harvey, 1974) and is written in the form

$$h_w = h_{i,j,k} - \frac{Q_{w[i,j,k]}}{2\pi T_{i,j}} \ln\,(r_e/r_w). \qquad (15)$$

Equation 15 assumes that: (1) flow is within a square well block and can be described by a steady-state equation with no source term except for the well discharge, (2) the aquifer is isotropic and homogeneous in the well block, (3) only one well is in the block and it fully penetrates the aquifer, (4) flow is laminar, and (5) well loss is negligible.

In an unconfined aquifer, the analogous equation is

$$H_w = \sqrt{H^2_{i,j,k} - \frac{Q_{w[i,j,k]}}{\pi K_{i,j}} \ln(r_e/r_w)} \qquad (16)$$

in which

$H_{i,j,k} = h_{i,j,k} -$ BOTTOM (I,J) is the saturated thickness of the aquifer at radius r_e (L);

H_w is the saturated thickness of the aquifer at the well (L);

$K_{i,j} = K_{xx[i,j]} = K_{yy[i,j]}$;

BOTTOM (I,J) = elevation of the bottom of the aquifer (The uppercase letters indicate that this parameter is identical to that used in the model.)

When the saturated thickness computed with equation 16 is negative, the message, 'X,Y WELL IS DRY' is generated. This situation has no effect on the computations, but should stimulate careful consideration of the value of results for subsequent time steps in the simulation.

The conditions when the Thiem equation or equation 16 will be accurate can be computed. Table 1 was prepared to give a few examples of the head values computed by the model with the Thiem equation for a well with a radius of 1.25 feet in an infinite leaky artesian aquifer and in an infinite nonleaky artesian aquifer. The analytic solutions for these conditions are included for comparison. A variable grid was used in the model but the dimensions of the well block were $\Delta x = \Delta y = 1{,}000$ feet. For conditions which depart significantly from the assumptions given above (for example, a well in a rectangular block with anisotropic transmissivity or a well in a large block that has a significant amount of leakage) the results using equations 15 and 16 should be checked with a more rigorous analysis. Additional drawdown due to the effects of partial penetration and well loss can be computed separately or added to the code as needed.

Table 1.—Comparison of drawdowns computed with equation 15 and the analytic values

Aquifer	Time step	Dimensionless time	Drawdown Approximation	Drawdown Analytic
Nonleaky		Tt/r^2S		
artesian -----	3	3.0×10^5	41.1	42.7
	14	3.7×10^7	58.3	58.1
Leaky		$K't/m^2S_s$		
artesian -----	3	0.028	51.8	52.1
	9	.44	57.1	57.3

Combined artesian–water-table simulation

Simulation of an aquifer that is partly confined and elsewhere has a free surface requires special computations for the transmissivity, storage coefficient, and leakage term. The following paragraphs describe the computations required. Some of the methods of coding these procedures have been adapted from Prickett and Lonnquist (1971).

Transmissivity

The transmissivity is computed as the saturated thickness of the aquifer times the hydraulic conductivity. This computation requires that the elevations of the top and bottom of the aquifer be specified. Where the aquifer crops out, the top of the aquifer is assigned a fictitious value greater than or equal to the elevation of the land surface.

Storage

The storage term requires special treatment at nodes where a conversion from artesian to water-table conditions, or vice versa, occurs during a time step. The program first checks for a change at a node during the last iteration. If there has been a change from artesian to water-table conditions, the storage term is

$$\frac{S_{y[i,j]}}{\Delta t}(h^n_{i,j,k}-h_{i,j,k-1})-\text{SUBS}$$

in which

$$\text{SUBS}=(h_{i,j,k-1}-\text{TOP(I,J)})\,(S_{i,j}-S_{y[i,j]})/\Delta t;$$

TOP(I,J) = elevation of the top of the aquifer.

The purpose of SUBS is to correctly apportion the storage coefficient and specific yield according to the relationship in figure 8a.

For a change from water-table to artesian conditions, the storage term is

$$\frac{S_{i,j}}{\Delta t}(h^n_{i,j,k}-h_{i,j,k-1})-\text{SUBS}$$

in which

$$\text{SUBS}=(h_{i,j,k-1}-TOP\text{(I,J)})\;(S_{y[i,j]}-S_{i,j})/\Delta t.$$

SUBS subtracts the storage coefficient and adds the specific yield for the distance *B* illustrated in figure 8b.

Leakage

To treat leakage more realistically if parts of an artesian aquifer change to water-table conditions, the maximum head difference across the confining bed is limited to $\hat{h}_{i,j,0}-\text{TOP (I,J)}$.

Two examples illustrate the calculation of leakage in conversion simulations. In figure 9a the head at the start of the pumping period, $h_{i,j,0}$ is below the water-table head, $\hat{h}_{i,j,0}$, but above the top of the aquifer; the current pumping level is below the top of the aquifer. The applicable equation is

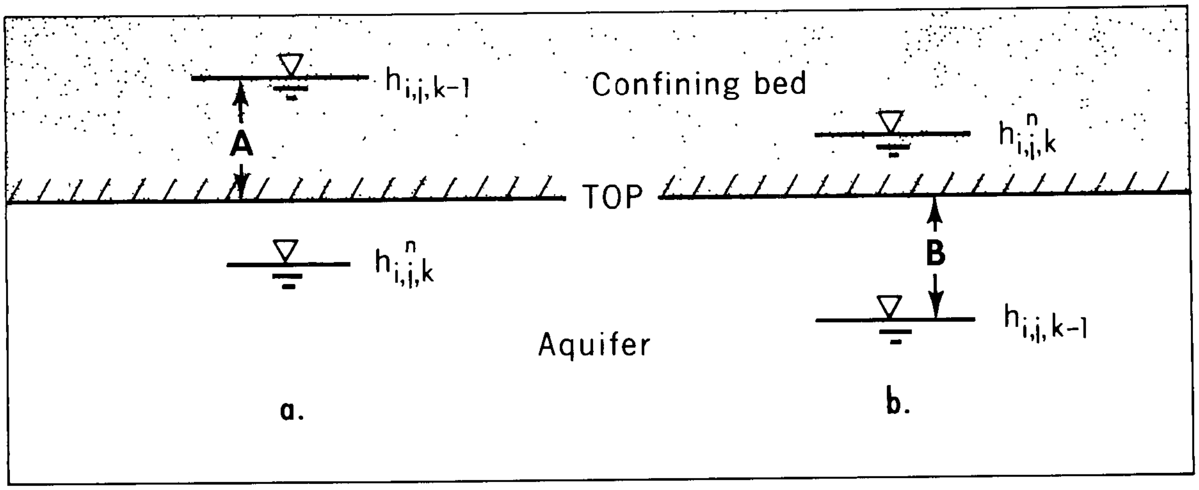


FIGURE 8.—Storage adjustment is applied to distance A in conversion from artesian to water-table conditions (a) and to distance B in conversion from water-table to artesian conditions (b).

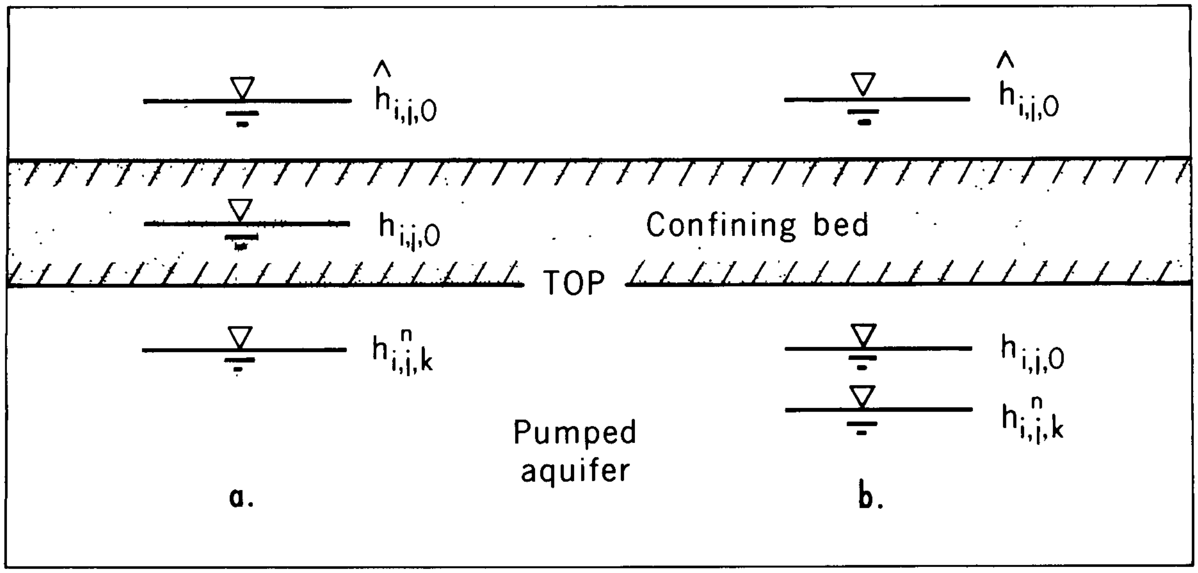


FIGURE 9.—Two of the possible situations in which leakage is restricted in artesian–water-table simulations.

$$q'_{i,j,k}=\frac{K'_{i,j}}{m_{i,j}}(\hat{h}_{i,j,0}-h_{i,j,0}) + T_L(h_{i,j,0}-\mathrm{TOP(I,J)}).$$

For this situation $q'_{i,j,k}$ appears on the right-hand side of the difference equation and is treated explicitly. Only if both $h_{i,j,0}$ and $h^{n}_{i,j,k}$ are above the top of the aquifer is the leakage term treated implicitly by including T_L in the E coefficient. This is accomplished in the code by setting $U=1$.

In the second example (fig. 9b), both $h_{i,j,0}$ and $h^{n}_{i,j,k}$ are below the top of the aquifer and the equation for leakage reduces to

$$q'_{i,j,k}=\frac{K'_{i,j}}{m_{i,j}}(\hat{h}_{i,j,0}-\mathrm{TOP(I,J)}).$$

If leakage across a subjacent confining bed is significant, it will be necessary to add a second leakage term. The flux described by this term will not be restricted where water-table conditions occur.

Test Problems

In a subsequent section the computational work required for solution of four test problems by the numerical techniques available in the model is analyzed. It is appropriate, however, to introduce the test problems here because they are used in the discussion of iteration parameters in the section on numerical techniques. The problems are for steady-state conditions since the resulting set of simultaneous equations are more difficult to solve than are the set of equations for transient problems which generally involve smaller head changes.

For each of these problems a closure criterion was chosen to decide when a solution is obtained to the set of finite-difference equations. (See Remson, Hornberger, and Molz, 1971, p. 185–186.) Normally, in this model, a solution is assumed if:

$$\mathrm{Max}\,|\,h^{n}-h^{n-1}\,|\leqq\varepsilon$$

where ε is an arbitrary closure criterion (L). For the purpose of the numerical comparisons given later in this documentation, the absolute value of the maximum residual (defined by equation 28) is used to compare methods.

The first problem is a square aquifer with uniform properties and grid spacing (fig. 10). The finite-different grid is 20×20, but only 18 rows and columns are inside the aquifer because the model requires that the first and last rows and columns be outside the aquifer boundaries. Two discharging wells and one recharge well are the stress on the system; boundaries are no flux except for part of one side which is a constant-head

PROBLEM CHARACTERISTICS

Transmissivity: Txx = Tyy = 0.1 ft²/s (0.009m²/s)
Grid spacing: Δx = Δy = 5000 ft (1500m)
Dimensions of grid. 18×18

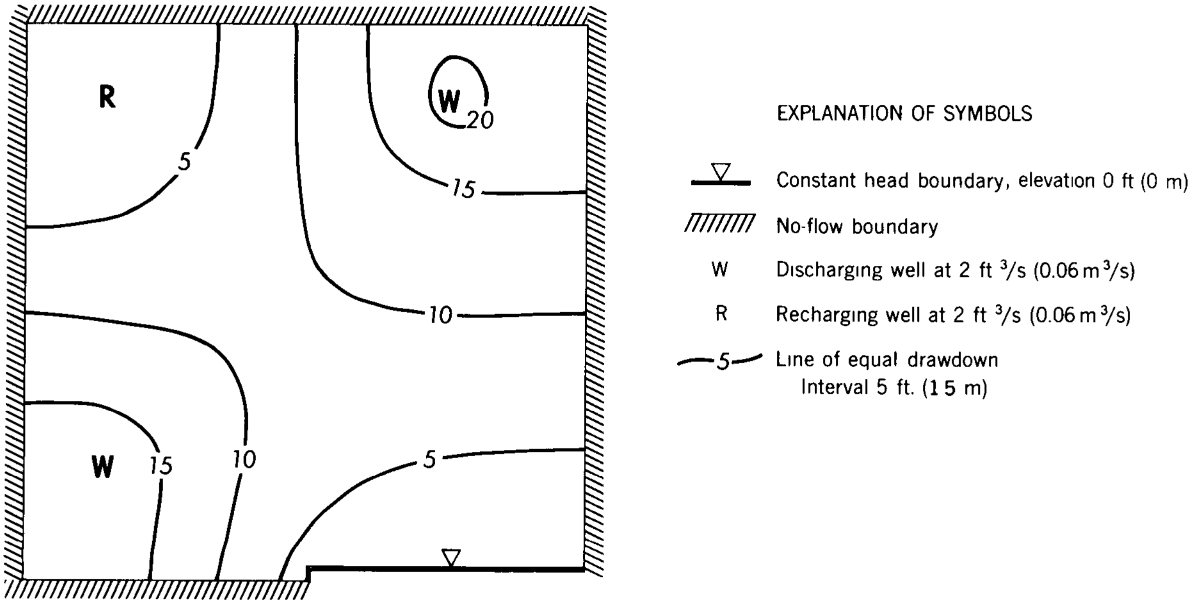


FIGURE 10.—Characteristics of test problem 1.

boundary. A closure criterion of 0.001 foot (0.0003 metre) was used.

Konikow (1974) designed the second problem in his analysis of ground-water pollution at the Rocky Mountain Arsenal northeast of Denver, Colo. It is included as one of the test problems because it is typical of many field problems and because there is some difficulty in obtaining a steady-state solution with the alternating-direction implicit procedure. The transmissivity distribution is shown in figure 11; note the extensive areas where the transmissivity is zero because the surficial deposits are unsaturated. The finite-difference grid representing this aquifer is 25×38 with square blocks 1,000 feet (300 metres) on a side. The model has constant-head boundaries at the South Platte River and where the aquifer extends beyond the limits of the model; elsewhere no-flux boundaries are employed. Although this is a water-table aquifer, it is assumed for problem 2 that transmissivity is independent of head. The model includes 49 irrigation wells and recharge from canals and irrigation. In figure 11 the observed water-table configuration is shown, and it is used as the initial surface for the simulation; the computed water table is generally within a few feet of the observed. For this problem the closure criterion is 0.001 foot (0.0003 metre).

The third problem is a cross-section with three horizontal layers and other characteristics shown in figure 12. Transmissivity equals hydraulic conductivity for this problem because it is conceived as a slice one unit wide. The values for transmissivity are arbitrary. Note in particular that the horizontal conductivity is 100 times the vertical conductivity in all layers and that the middle layer acts as a confining layer between the upper and lower layers. The coefficients $B_{i,j}$ and $H_{i,j}$, however, are 100 times greater than the horizontal coefficients $D_{i,j}$ and $F_{i,j}$ because of

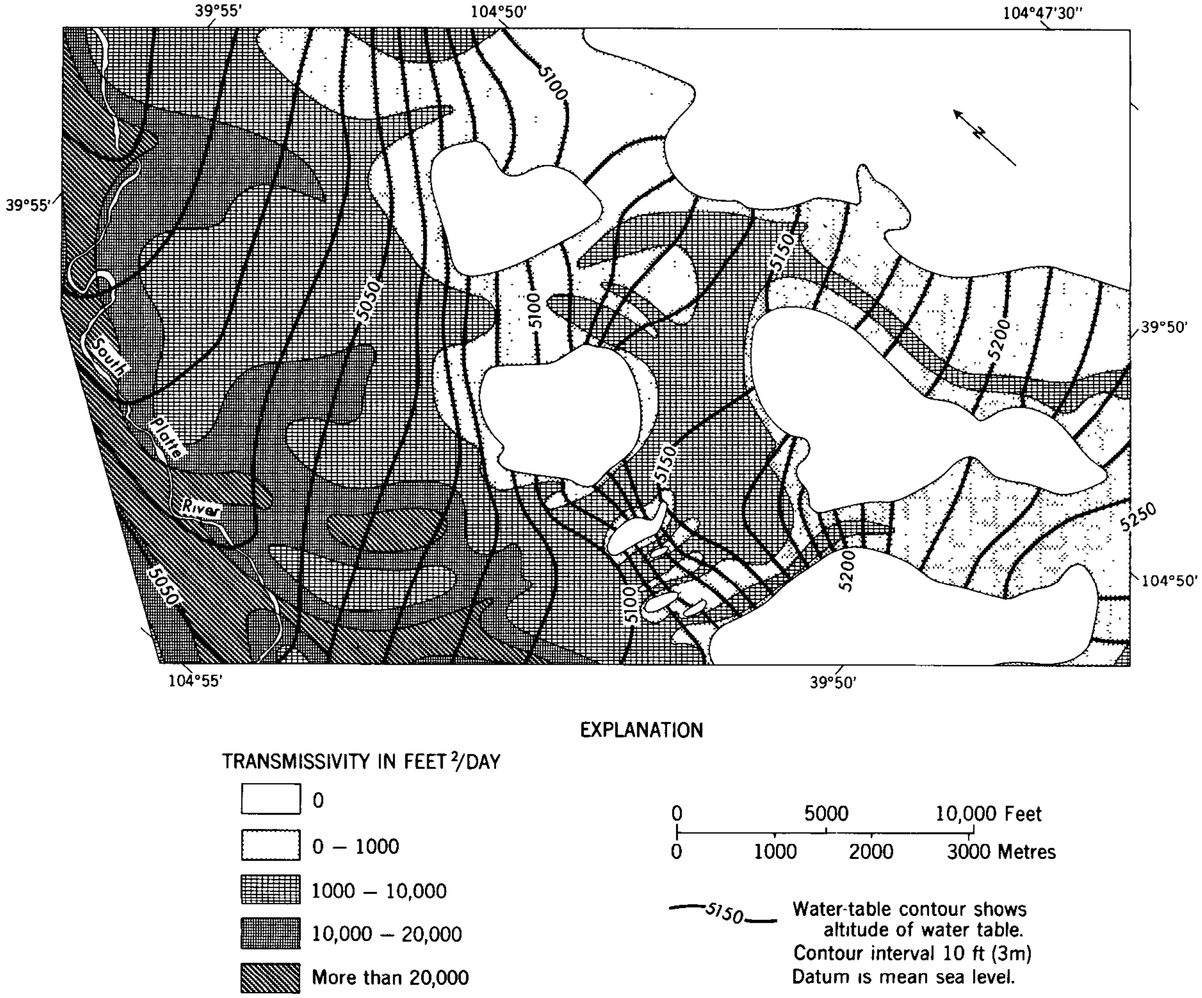


FIGURE 11.—Transmissivity and observed water-table configuration for test problem 2 (fieldwork and model design by Konikow, 1975).

the grid spacing used. For this problem, the closure criterion is 0.0001 feet (0.00003 metre).

In the third problem the upper boundary (the water table) is fixed as a constant-head boundary. It could also be treated as a no-flow boundary which would effectively confine the system. This model was not designed specifically for simulation of cross sections, and consequently it does not have provision for a moving boundary. Rather than modifying this one-phase model for a moving-boundary problem, it would be better to design a model specifically for this purpose. The two-phase model described by Freeze (1971) is a good example.

The fourth problem is to consider the water-table case of the second problem. The only difference from problem 2 is that transmissivity is dependent upon (1) head in the aquifer, (2) aquifer base elevation, and (3) hydraulic conductivity of the aquifer.

Numerical Solution

In Pinder (1969) and Trescott (1973) the iterative, alternating-direction implicit procedure (ADI) was the only option available for numerical solution. For many field problems ADI is convergent and competitive, in terms of the computational work required,

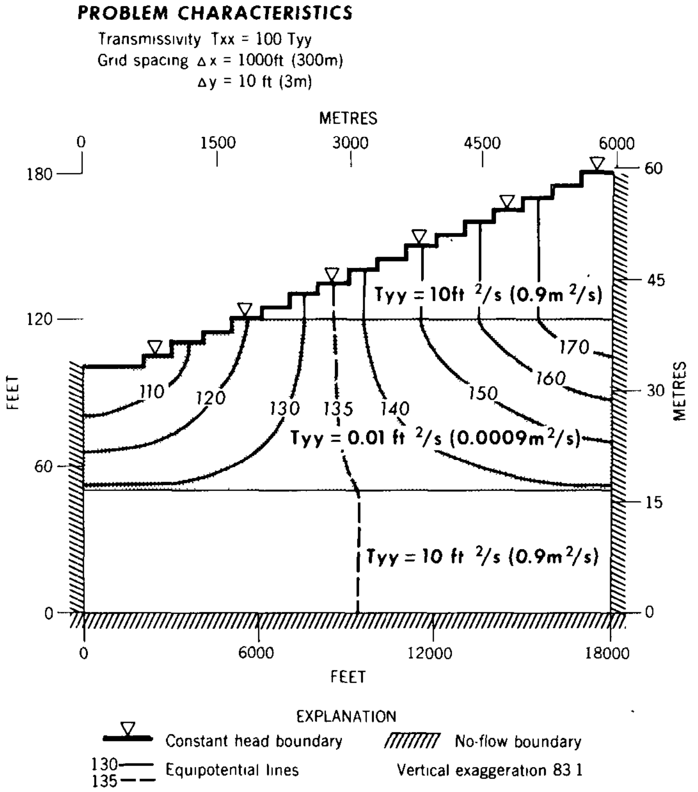


FIGURE 12.—Characteristics of test problem 3.

with other iterative techniques available. It may be difficult, however, to obtain a solution for some problems with ADI (for example, steady-state simulations involving extremely variable coefficients). Consequently, it is convenient to have available other numerical techniques that may be more suited than ADI to particular problems. The three numerical methods available with this model are ADI, the strongly implicit procedure (SIP), and line successive overrelaxation (LSOR).

The following sections outline the computational algorithms for the three numerical methods. More details are given in the discussion on SIP, because that method is more complex.

For additional details on the theory behind the methods and rigorous analysis of convergence rates, see for example, Varga (1962) and Remson, Hornberger, and Molz (1971). The methods are presented in order of increasing complexity. In general, the more complex methods converge more rapidly and are applicable to more types of problems than the simpler methods such as LSOR. For clarity, the numerical treatment of the source term is left to other sections.

Line successive overrelaxation

Line successive overrelaxation (LSOR) improves head values one row (or column) at a time. Whether the solution is oriented along rows or columns is generally immaterial for isotropic problems but has a significant affect on the convergence rate in anisotropic problems. The solution should be oriented in the direction of the larger coefficients, either $B_{i,j}$ and $H_{i,j}$ or $D_{i,j}$ and $F_{i,j}$ (Breitenbach, Thurnau, and van Poollen, 1969, p. 159). Differences in the magnitude of the coefficients may result from anisotropic transmissivity or from a large difference in grid spacing between the x and y directions. In problem 3 the largest transmissivity is in the horizontal direction in each layer, but the small grid spacing in the vertical direction makes the coefficients $B_{i,j}$ and $H_{i,j} \gg D_{i,j}$ and $F_{i,j}$.

With the solution oriented along rows, an intermediate value is computed by the line Gauss-Seidel iteration formula,

$$Dh^{\dagger}_{j-1}+Eh^{\dagger}+Fh^{\dagger}_{j+1}=Q_{\lambda},\ j=1,2,\ldots,N_x \quad (17a)$$

in which

$$Q_{\lambda}=W-Bh^{n}_{i-1}-Hh^{n-1}_{i+1}-\frac{S}{\Delta t}h_{k-1};$$

$h^{\dagger}$ is the intermediate head value at node (i,j);

N_x is the number of nodes in a row.

Equation 17a can be expressed in matrix form as

$$\bar{\bar{A}}_{\lambda}\bar{h}^{\dagger}=\bar{Q}_{\lambda}. \quad (17b)$$

In order to reduce rounding errors, equation 17b is put in residual form. (See Weinstein, Stone, and Kwan, 1969, p. 283, and Breitenbach, Thurnau, and van Poollen, 1969, p. 159.) This is accomplished by adding and subtracting $\bar{\bar{A}}_{\lambda}\bar{h}^{n-1}$ to the right-hand side of equation 17b giving

$$\bar{\bar{A}}_{\lambda}\bar{h}^{\dagger}=\bar{Q}_{\lambda}+\bar{\bar{A}}_{\lambda}\bar{h}^{n-1}-\bar{\bar{A}}_{\lambda}\bar{h}^{n-1}. \quad (17c)$$

Rearrange equation 17c to read

$$\bar{\bar{A}}_{\lambda}\bar{\xi}^{\dagger}=\bar{R}_{\lambda}{}^{n-1} \quad (17d)$$

in which

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

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

Equation 17d is the LSOR residual formulation and expanded has the following form for a 3×3 problem (fig. 13):

$$\begin{array}{r}\text{row 1}\left\{\begin{array}{l}\\ \\ \\ \end{array}\right.\\ \\ \\ \\ \\ \\ \end{array}\begin{bmatrix} E_1 & F_1 & & & & & & & \\ D_2 & E_2 & F_2 & & & & & & \\ & D_3 & E_3 & 0 & & & & & \\ & & 0 & E_4 & F_4 & & & & \\ & & & D_5 & E_5 & F_5 & & & \\ & & & & D_6 & E_6 & 0 & & \\ & & & & & 0 & E_7 & F_7 & \\ & & & & & & D_8 & E_8 & F_8 \\ & & & & & & & D_9 & E_9 \end{bmatrix}\begin{bmatrix}\xi^{\dagger}_1\\ \xi^{\dagger}_2\\ \xi^{\dagger}_3\\ \xi^{\dagger}_4\\ \xi^{\dagger}_5\\ \xi^{\dagger}_6\\ \xi^{\dagger}_7\\ \xi^{\dagger}_8\\ \xi^{\dagger}_9\end{bmatrix}=\begin{bmatrix}R^{n-1}_{\lambda 1}\\ R^{n-1}_{\lambda 2}\\ R^{n-1}_{\lambda 3}\\ R^{n-1}_{\lambda 4}\\ R^{n-1}_{\lambda 5}\\ R^{n-1}_{\lambda 6}\\ R^{n-1}_{\lambda 7}\\ R^{n-1}_{\lambda 8}\\ R^{n-1}_{\lambda 9}\end{bmatrix}$$

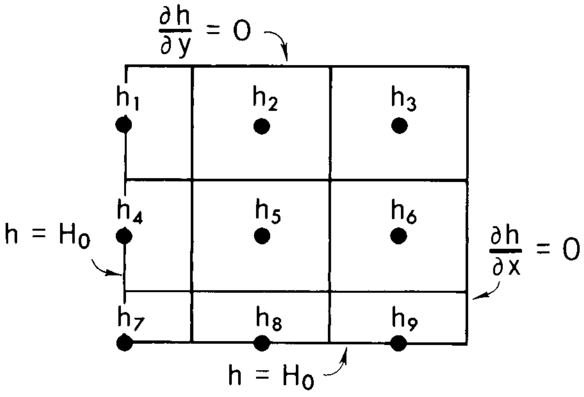


FIGURE 13.—Hypothetical problem with 9 interior nodes.

Boundary conditions are not included in this equation because they are treated in the model without adding or subtracting terms to $\bar{R}^{n-1}_{\lambda}$.

The first row is solved by the Thomas algorithm for simultaneous equations with a tridiagonal coefficient matrix. The Thomas algorithm is given in many references. (For example, see Pinder and Bredehoeft, 1968; von Rosenberg, 1969; Remson, Hornberger, and Molz, 1971.) It is outlined below for equation 17 using notation from the program code (The coefficients D, E, F, and the known term $\bar{R}^{n-1}$ have been subscripted with $[i,j]$ for clarity). BE_j is an intermediate coefficient.

Recognizing that

$$D_{i,1}=F_{i,N_x}=0,$$

an intermediate vector $\bar{G}$ is computed by forward substitution as

$$\begin{aligned} W &= E_{i,j}-D_{i,j}(BE_{j-1}),\\ BE_j &= F_{i,j}/W\\ G_j &= (R^{n-1}{}_{[i,j]}-D_{i,j}(G_{j-1}))/W. \end{aligned}$$

The values of $\bar{\xi}^{\dagger}$ for row i are then computed by backward substitution as

$$\xi^{\dagger}_{i,j,k}=G_{j}-BE_{j}\xi^{\dagger}_{i,j+1,k}$$

where

$$\xi^{\dagger}_{i,N_x,k}=G_{N_x}$$

since

$$BE_{N_x}=0.$$

The head values for row 1 are then computed by the equation

$$h^n_{i,j}=h^{n-1}_{i,j}+\omega\xi^{\dagger}_{i,j},\ j=1,\ldots,N_x$$

If ω is 1, the solution is by the line Gauss-Seidel formula, but convergence is slow in general. The convergence rate is improved significantly by "overrelaxation" with $1<\omega<2$. Discussion of the acceleration parameter is deferred until after the following section on two-dimensional correction.

Two-dimensional correction to LSOR

In certain problems, the rate of convergence of LSOR can be improved by applying a one-dimensional correction (1DC) procedure introduced by Watts (1971) or the extended two-dimensional correction (2DC) method described by Aziz and Settari (1972). These methods remove the components of certain eigenvectors in the LSOR iteration matrix from the solution vector. If the eigenvalues associated with these eigenvectors dominate the problem, particularly those including anisotropy, the convergence rate is greatly improved.

The 2DC method is applied after one or more LSOR iterations. The corrected head values are used as an improved starting point for the next iteration and the process is repeated until convergence is achieved.

The two-dimensional correction for the head at (i,j) is defined as

$$h^{n*}_{i,j,k}=h^n_{i,j,k}+\alpha_i+\hat{\beta}_j,\quad i=1,\ldots,N_y,\ j=1,\ldots,N_x$$

in which

$h^{n*}_{i,j,k}$ is the corrected head at iteration n;
α_i is the correction for row i;
$\hat{\beta}_j$ is the correction for column j.
N_y is the number of nodes in a column.

An approximate equation for $\bar{\alpha}$ is

$$B'_i\alpha_{i-1}+E'_i\alpha_i+H'_i\alpha_{i+1}=R'_i,\ i=1,2,\ldots,N_y \quad (18)$$

in which

$$B'_i=-\sum_j B_{i,j};$$

$$E'_i=\sum_j\left(B_{i,j}+H_{i,j}+\frac{S_{i,j}}{\Delta t}\right);$$

$$H'_i=-\sum_j H_{i,j};$$

$$R'_i=\sum_j R^n_{i,j};$$

$$R^n_{i,j}=B_{i,j}h^n_{i-1,j,k}+D_{i,j}h^n_{i,j-1k}+E_{i,j}h^n_{i,j,k}+F_{i,j}h^n_{i,j+1,k}+H_{i,j}h^n_{i+1,j,k}+\frac{S_{i,j}}{\Delta t}h_{i,j,k-1}-W_{i,j,k};$$

An approximate equation for $\bar{\hat{\beta}}$ is

$$D'_j\hat{\beta}_{j-1}+E'_j\hat{\beta}_j+F'_j\hat{\beta}_{j+1}=R'_j, j=1,2,\ldots,N_x \quad (19)$$

in which

$$D'_j=-\sum_i D_{i,j};$$

$$E'_j=\sum_i\left(D_{i,j}+F_{i,j}+\frac{S_{i,j}}{\Delta t}\right);$$

$$F'_j=-\sum_i F_{i,j};$$

$$R'_j=\sum_i R^n_{i,j}$$

Equations 18 and 19 are derived with the following equations

$$\sum_{j=1}^{N_x} R^{n*}_{i,j}=0,\ i=1,2,\ldots,N_y$$

and

$$\sum_{i=1}^{N_y} R^{n*}_{i,j}=0,\ j=1,2,\ldots,N_x$$

which force the sum of residuals for each row and each column to zero when the vector $\bar{h}^{n*}$ is substituted into equation 8. Aziz and Settari (1972) give the exact equations for $\bar{\alpha}$ and $\bar{\beta}$ but point out that equations 18 and 19 are good approximations and, in practice, are easier to solve. For example, equation 19, which used alone is Watts' 1DC method, is written in matrix form as

$$\begin{bmatrix} E'_1 & F'_1 & \\ D'_2 & E'_2 & F'_2 \\ & D'_3 & E'_3 \end{bmatrix} \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = \begin{bmatrix} R'_1 \\ R'_2 \\ R'_3 \end{bmatrix}$$

for the problem in figure 13. Equation 18 has an analogous form and both are easily solved by the Thomas algorithm.

Note that $\bar{\alpha}$ and $\bar{\beta}$ in the model are zero for those rows and columns in which one or more constant-head nodes are located. If $\bar{\alpha}$ and $\bar{\beta}$ were not zero it would not be possible to maintain a constant value at the appropriate nodes. As Watts (1973) points out, therefore, the procedure is most useful in simulations dominated by no-flow boundaries. For those simulations in which 2DC is useful, it is generally better to apply the corrections after several rather than after each LSOR iteration. After experimenting with a few problems, we have found it practical to apply 2DC after every 5 LSOR iterations.

LSOR acceleration parameter

The optimum value of ω for maximum rate of convergence lies between 1 and 2 and is commonly between 1.6 and 1.9. If only one or two runs will be made on a problem, it is probably best to choose an ω based on experience. If many runs will be made, it will be worthwhile to use an ω close to the optimum value. For simple problems ω_{opt} can be computed as explained, for example, by Remson, Hornberger, and Molz (1971, p. 188–199) using the equation

$$\omega = \frac{2}{1+\sqrt{1-\rho(G)}} \tag{20}$$

in which

$$\rho(G) \cong \frac{\left|\xi^{\dagger(n)}_{\max}\right|}{\left|\xi^{\dagger(n-1)}_{\max}\right|}$$

$\rho(G)$ is the spectral radius (dominant eigenvalue) of the Gauss-Seidel iteration matrix. For typical field problems it is possible to use equation 20 to estimate ω_{opt} in an iterative process if 2DC is not used. In the first simulation of the problem, set $\omega = 1.0$ and allow at least 100 iterations. In applying this method to problems 1, 2, and 3 it took 25 iterations to arrive at ω_{opt} for problem 2, but about 100 iterations to obtain ω_{opt} for problem 1 and 3. Obviously this method may involve a lot of computational effort to obtain ω_{opt}. More efficient methods using equation 20 have been devised to update ω during the iteration process. For example, Breitenbach, Thurnau, and van Poollen (1969) use a modified form of Varga's (1962) "power method," Carré's (1961) method is described by Remson, Hornberger, and Molz (1971, p. 199–203), and Cooley (1974) has a simple method for improving ω for transient problems.

Figure 14 illustrates the rate of convergence of LSOR and LSOR+2DC for test problems 1, 2, and 3 using different acceleration parameters chosen by trial and error. The values exceeding 100 iterations for problem 1 were estimated by using a plot, which is nearly a straight line, of the absolute value of the log of the maximum residual (defined by equation 28) versus the number of iterations. This plot was extrapolated to the value of maximum residual that corresponded roughly to the closure criterion chosen for the problem. The same procedure was used on problem 3 for values exceeding 200 iterations.

For problem 1 the optimum acceleration parameter is 1.87 for LSOR. Two-dimensional correction significantly improves the convergence rate of LSOR for this problem with an optimum acceleration parameter of 1.7. In problem 2, 2DC had no effect on the rate of convergence of LSOR because of the numerous constant-head nodes in the problem. Consequently, the optimum acceleration parameter is 1.6 with or without the application of 2DC. In problem 3, with LSOR oriented across the bedding, ω_{opt} is 1.88 for LSOR and about 1.70 for LSOR+2DC. Note in problems 1 and 3 that finding ω_{opt} for LSOR is more critical than with LSOR+2DC. LSOR is poorly suited for problem 4 because too many nodes drop out in the iteration process if $1<\omega<2$. Satisfactory results for problem 4 at the expense of slow convergence are obtained if $\omega = 0.5$ (See fig. 23.)

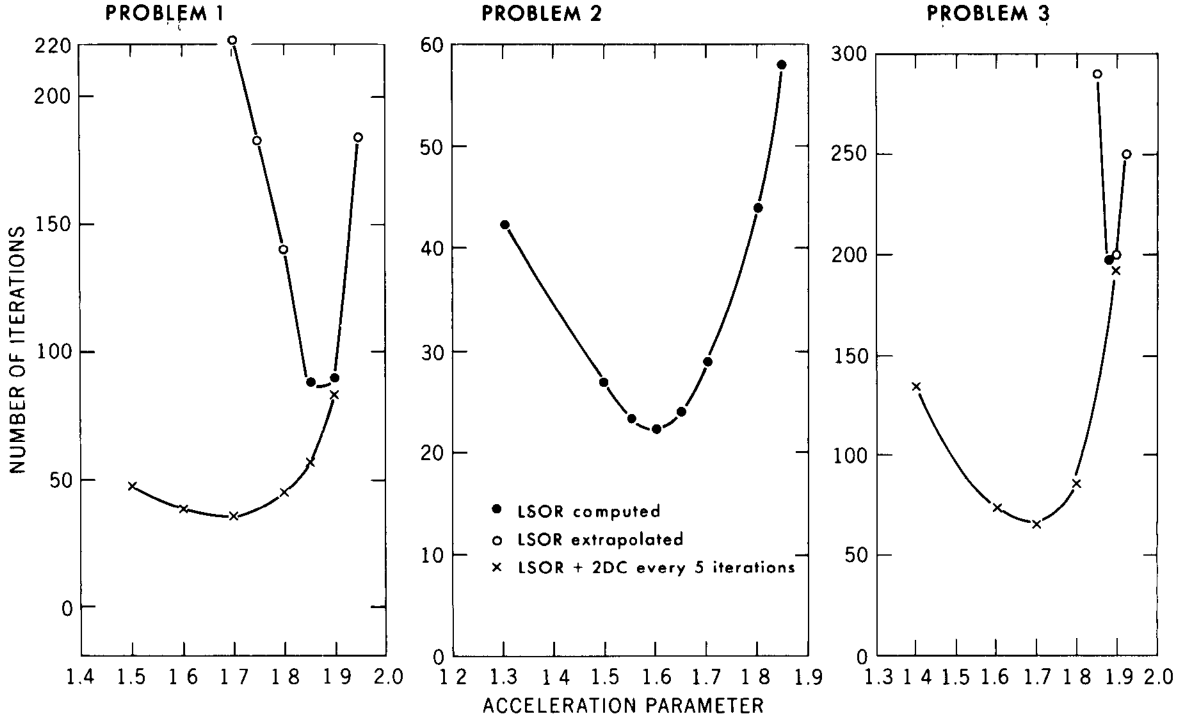


FIGURE 14.—Number of iterations required for solution by LSOR and LSOR + 2DC using different acceleration parameters.

Alternating-direction implicit procedure

Peaceman and Rachford (1955) described the iterative, alternating-direction implicit procedure for solution of a steady-state (Laplace) equation in two space dimensions. This procedure, however, is equally applicable to transient problems where it has the advantage of allowing larger time steps than can be used with non-iterative ADI. (Non-iterative ADI was used by Pinder and Bredehoeft, 1968.) In the ADI technique, two sets of matrix equations are solved each iteration. The equations for rows in which head values along rows are computed implicitly and those along columns are obtained from the previous column computations are defined as

$$Dh^{n-\frac{1}{2}}_{j-1} + E_r h^{n-\frac{1}{2}} + F h^{n-\frac{1}{2}}_{j+1} = Q_r, \; j=1,2,\ldots,N_x \quad (21a)$$

in which

$$E_r = -\left(D+F+\frac{S}{\Delta t}+M_l\right);$$

$$Q_r = -Bh^{n-1}_{i-1} + (B+H-M_l)\,h^{n-1} - Hh^{n-1}_{i+1} - \frac{S}{\Delta t}h_{k-1} + W;$$

M_l is the iteration parameter;

l is the iteration parameter index.

In matrix form equation 21a is

$$\bar{\bar{A}}_r\bar{h}^{n-\frac{1}{2}} = \bar{Q}_r. \quad (21b)$$

To put equation 21b in residual form, add and subtract $\bar{\bar{A}}_r\bar{h}^{n-1}$ to the right-hand side giving

$$\bar{\bar{A}}_r\bar{h}^{n-\frac{1}{2}} = \bar{Q}_r - \bar{\bar{A}}_r\bar{h}^{n-1} + \bar{\bar{A}}_r\bar{h}^{n-1} \quad (21c)$$

Rearrange equation 21c to read:

$$\bar{\bar{A}}_r\bar{\xi}^{n-\frac{1}{2}} = \bar{R}_r^{\,n-1} \quad (21d)$$

in which

$$\bar{\xi}^{n-\frac{1}{2}} = \bar{h}^{n-\frac{1}{2}} - \bar{h}^{n-1};$$

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

Equation 21d is the ADI row formula in residual form. Its matrix form is the same as that for equation 17d and is solved for each row by the Thomas algorithm. To complete the first half of the ADI iteration, $\bar{h}^{n-\frac{1}{2}}$ is computed by

$$\bar{h}^{n-\frac{1}{2}}=\bar{h}^{n-1}+\bar{\xi}^{n-\frac{1}{2}}.$$

The equations in which head values along columns are considered implicitly and those along rows explicitly are written as:

$$Bh^n_{i-1}+E_c h^n+Hh^n_{i+1}=Q_c,\ i=1,2,\ldots,N_y \qquad (22a)$$

in which

$$E_c=-\left(B+H+\frac{S}{\Delta t}+M_l\right);$$

$$Q_c=-Dh^{n-\frac{1}{2}}_{j-1}+(D+F-M_l)\,h^{n-\frac{1}{2}}$$

$$-Fh^{n-\frac{1}{2}}_{j+1}-\frac{S}{\Delta t}h_{k-1}+W.$$

Equation 22a in matrix form is

$$\bar{\bar{A}}_c\bar{h}^n=\bar{Q}_c. \qquad (22b)$$

By adding and subtracting $\bar{\bar{A}}_c\bar{h}^{n-\frac{1}{2}}$ to the right-hand side of equation 22b, it can be put in the residual form

$$\bar{\bar{A}}_c\bar{\xi}^n=\bar{R}_c^{n-\frac{1}{2}}; \qquad (22c)$$

in which

$$\bar{\xi}^n=\bar{h}^n-\bar{h}^{n-\frac{1}{2}};$$

$$\bar{R}_c^{n-\frac{1}{2}}=\bar{Q}_c-\bar{\bar{A}}_c\bar{h}^{n-\frac{1}{2}}.$$

Equation 22c is solved for each column by the Thomas algorithm, and the vector $\bar{h}^n$ for each row is obtained by the equation

$$\bar{h}^n=\bar{h}^{n-\frac{1}{2}}+\bar{\xi}^n.$$

A set of iteration parameters is computed by the equation

$$M_l=\omega_l\,(B+D+F+H)$$

in which ω ranges between a minimum defined by

$$\omega_{\min}=\underset{\text{(over grid)}}{\text{Min}}\left[\frac{\pi^2}{2N_x^2}\,\frac{1}{1+\left(\dfrac{T_{yy\,[i,j]}\,(\Delta x_j)^2}{T_{xx\,[i,j]}\,(\Delta y_i)^2}\right)},\ \frac{\pi^2}{2N_y^2}\,\frac{1}{1+\left(\dfrac{T_{xx\,[i,j]}\,(\Delta y_i)^2}{T_{yy\,[i,j]}\,(\Delta x_j)^2}\right)}\right] \qquad (23a)$$

and a maximum given by

$$\omega_{\max}=\begin{cases}1 & [T_{xx}\cong T_{yy}];\\ 2 & [T_{xx}>>T_{yy}\text{ or }T_{yy}>>T_{xx}].\end{cases}$$

The set of parameters are spaced in a geometric sequence given by

$$\omega_{l+1}=\gamma\omega_l \qquad (23b)$$

in which

$$\ln\gamma=\frac{\ln(\omega_{\max}/\omega_{\min})}{L-1}. \qquad (23c)$$

L = the number of iteration parameters used.

The iteration parameters starting with $\omega_{\min}$ are cycled until convergence is achieved.

Equation 23a is based on a von Neuman error analysis of the normalized flow equations. (See, for example, Weinstein, Stone, and Kwan, 1969.) It will compute the optimum $\omega_{\min}$ only for simple problems. For general problems $\omega_{\min}$ computed by equation 23a may or may not be close to the optimum $\omega_{\min}$ for the problem. This is illustrated in figure 15 in which the rate of reduction in the maximum residual for arbitrarily chosen minimum parameters is compared with that for $\omega_{\min}$ computed with equation 23a. Ten parameters were used in problems 1 and 2, and four parameters were used in problem 3. The lines on figure 15 are meant to show the general trend only. The convergence rate using the best $\omega_{\min}$ in figure 15 is nearly the same as that computed with equation 23a for problem 1, but there is a significant difference in rates for problems 2 and 3. (See figs. 21 and 22.)

The other factor that may be critical in determining the rate of convergence using ADI is the number of parameters. In general, the number of parameters is chosen as 5 if $\omega_{\max}-\omega_{\min}$ is about two orders of magnitude; if $\omega_{\max}-\omega_{\min}$ is three or more orders of magnitude, 7 or more parameters are chosen.

For the test problems, the number of iteration parameters were varied from 4 to 10 (fig. 16). The minimum parameter was calculated by equation 23a; the maximum parameter was 1 for problems 1 and 2 and was 2 for problem 3. The number of parameters had a relatively small effect in determining the rate of convergence for problems 1 and 3. For problem 2, however, the computations do not converge using 4 or 5 parameters. Problem 2 can be solved with ADI using 6 to 10 parameters with 10 parameters giving the most rapid convergence. ADI did not give satisfactory solutions for problem 4 (an excessive number of nodes always drop out of the solution) and, consequently, no results for problem 4 are shown in figure 16.

When difficulties occur with ADI in steady-state simulations, rather than experimenting with the critical minimum parameter or the number of parameters, it may be worthwhile to make the simulation a transient problem. In effect, $S/\Delta t$ is used as an additional iteration parameter. If the storage coefficient is not made too large or the time step too small, steady state should be achieved within a reasonable number of time steps with rapid convergence at each time step.

Strongly implicit procedure

The set of equations (corresponding to equation 8) for the 3×3 problem in figure 13 may be expressed in matrix form as

$$\bar{\bar{A}}\;\bar{h}=\bar{Q} \qquad (24)$$

$$\begin{array}{c}N_x+1\text{ elements}\\ \left.\begin{array}{c}\\ \\ \\ \\ \end{array}\right\}\,N_x+1\text{ elements}\end{array}\quad\left[\begin{array}{ccccccccc}
E_1 & F_1 & 0 & H_1 & & & & & \\
D_2 & E_2 & F_2 & 0 & H_2 & & & & \\
0 & D_3 & E_3 & 0 & 0 & H_3 & & & \\
B_4 & 0 & 0 & E_4 & F_4 & 0 & H_4 & & \\
 & B_5 & 0 & D_5 & E_5 & F_5 & 0 & H_5 & \\
 & & B_6 & 0 & D_6 & E_6 & 0 & 0 & H_6 \\
 & & & B_7 & 0 & 0 & E_7 & F_7 & 0 \\
 & & & & B_8 & 0 & D_8 & E_8 & F_8 \\
 & & & & & B_9 & 0 & D_9 & E_9
\end{array}\right]\left[\begin{array}{c}h_1\\h_2\\h_3\\h_4\\h_5\\h_6\\h_7\\h_8\\h_9\end{array}\right]=\left[\begin{array}{c}Q_1\\Q_2\\Q_3\\Q_4\\Q_5\\Q_6\\Q\\Q_8\\Q_9\end{array}\right]$$

Direct solution of equation 24 by Gaussian elimination usually requires more work and computer storage than iterative methods for problems of practical size because $\bar{\bar{A}}$ decomposes into a lower triangular matrix with non-zero elements from B to E in each row and an upper triangular matrix with non-zero elements from E to H in each row. All of these intermediate coefficients must be computed during Gaussian elimination, and the coefficients in the upper triangular matrix must be saved for backward substitution.

To reduce the computation time and storage requirements of direct Gaussian elimination, Stone (1968) developed an iterative method using approximate factorization. In this approach a modifying matrix $\bar{\bar{B}}$ is added to $\bar{\bar{A}}$ forming $(\overline{\overline{A+B}})$ so that equation 24 becomes

$$(\overline{\overline{A+B}})\bar{h}=\bar{Q}+\bar{\bar{B}}\bar{h}. \qquad (25)$$

$(\overline{\overline{A+B}})$ can be made close to $\bar{\bar{A}}$ but can be factored into the product of a lower triangular matrix $\bar{\bar{L}}$ and an upper triangular matrix $\bar{\bar{U}}$, each of which has no more than three non-zero elements in each row, regardless of the size of N_x and N_y. Therefore, if the right-hand side of equation 25 is known, simple recursion formulas can be derived, resulting in a considerable savings in computer time and storage. This leads to the iteration scheme

$$(\overline{\overline{A+B}})\bar{h}^n=\bar{Q}+\bar{\bar{B}}\bar{h}^{n-1}. \qquad (26)$$

In order to transform equation 26 into a residual form, $\bar{\bar{A}}\bar{h}^{n-1}$ is subtracted from both sides giving

$$(\overline{\overline{A+B}})\bar{\xi}^n=\bar{R}^{n-1} \qquad (27)$$

in which

$$\begin{aligned}\bar{\xi}^n &=\bar{h}^n-\bar{h}^{n-1};\\ \bar{R}^{n-1} &=\bar{Q}-\bar{\bar{A}}\bar{h}^{n-1}.\end{aligned} \qquad (28)$$

The iterative scheme defined by equation 26 or 27 is closer to direct methods of solution (more implicit) than ADI (hence the term strongly implicit procedure or SIP). The SIP algorithm requires (1) relationships among the elements of $\bar{\bar{L}}$, $\bar{\bar{U}}$ and $(\overline{\overline{A+B}})$ defined by rules of matrix multiplication for the equation

$$\bar{\bar{L}}\;\bar{\bar{U}}=(\overline{\overline{A+B}}), \qquad (29)$$

and (2) relationships among the elements of $\bar{\bar{A}}$ and $(\overline{\overline{A+B}})$.

$\bar{\bar{L}}$ and $\bar{\bar{U}}$ have the following form for a general 3×3 problem (much of the notation is adapted from Remson, Hornberger, and Molz, 1971);

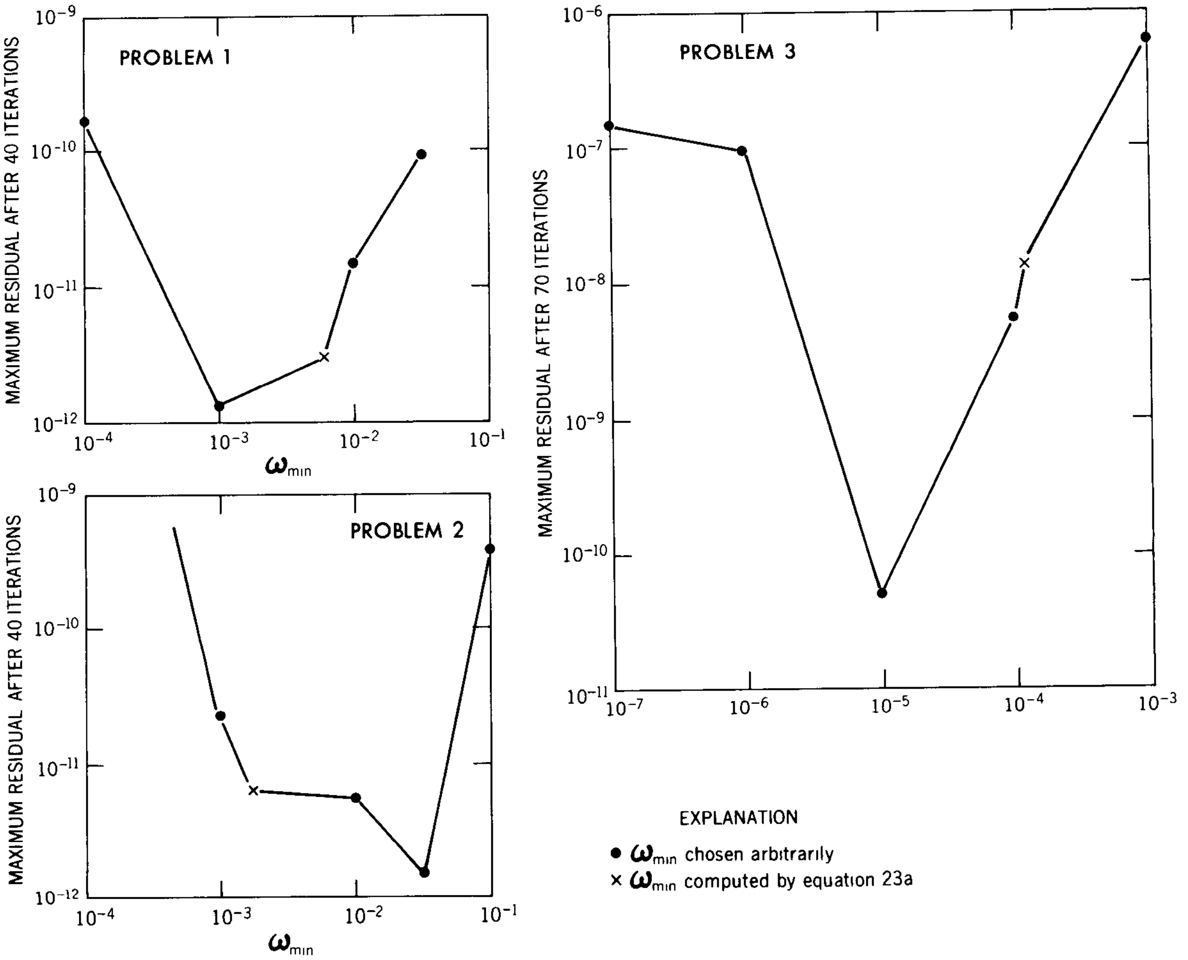


FIGURE 15.—Reduction in the maximum residual for problems 1 to 3 for selected ω_{min} used to compute the ADI parameters.

N_x Elements

$$\bar{\bar{L}}=\begin{bmatrix}
\gamma_1 &&&&&&&& \\
\beta_2 & \gamma_2 &&&&&&& \\
0 & \beta_3 & \gamma_3 &&&&&& \\
\alpha_4 & 0 & \beta_4 & \gamma_4 &&&&& \\
 & \alpha_5 & 0 & \beta_5 & \gamma_5 &&&& \\
 && \alpha_6 & 0 & \beta_6 & \gamma_6 &&& \\
 &&& \alpha_7 & 0 & \beta_7 & \gamma_7 && \\
 &&&& \alpha_8 & 0 & \beta_8 & \gamma_8 & \\
 &&&&& \alpha_9 & 0 & \beta_9 & \gamma_9
\end{bmatrix}$$

N_x Elements

$$\bar{\bar{U}}=\begin{bmatrix}
1 & \delta_1 & 0 & \eta_1 &&&&& \\
 & 1 & \delta_2 & 0 & \eta_2 &&&& \\
 && 1 & \delta_3 & 0 & \eta_3 &&& \\
 &&& 1 & \delta_4 & 0 & \eta_4 && \\
 &&&& 1 & \delta_5 & 0 & \eta_5 & \\
 &&&&& 1 & \delta_6 & 0 & \eta_6 \\
 &&&&&& 1 & \delta_7 & 0 \\
 &&&&&&& 1 & \delta_8 \\
 &&&&&&&&
\end{bmatrix}$$

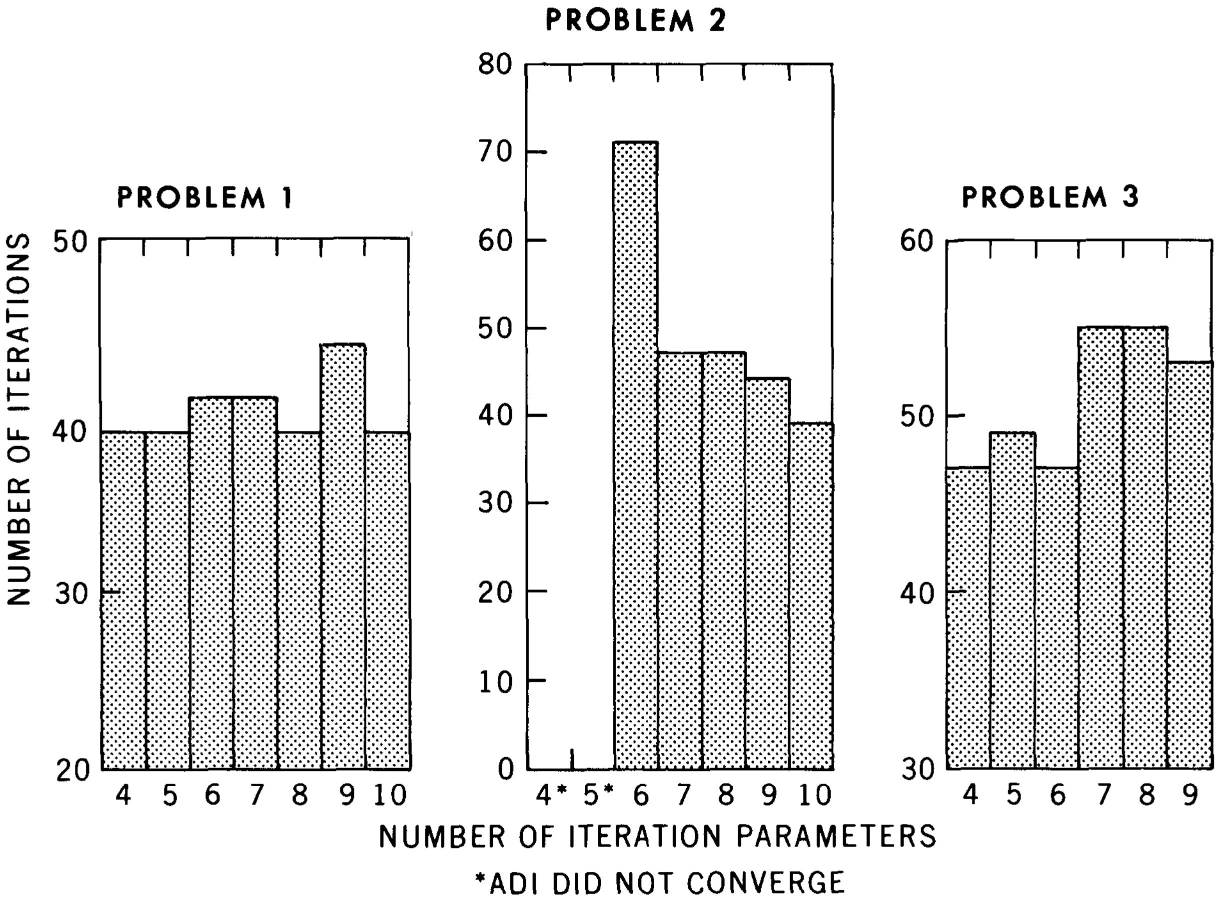


FIGURE 16.—Number of iterations required for solution of the test problems with ADI using different numbers of parameters.

The product $\bar{\bar{L}}\bar{\bar{U}} = (\overline{\overline{A+B}})$ is

$$
(\overline{\overline{A+B}}) = \begin{bmatrix}
\hat{E}_1 & \hat{F}_1 & 0 & \hat{H}_1 & & & & & \\
\hat{D}_2 & \hat{E}_2 & \hat{F}_2 & \hat{G}_2 & \hat{H}_2 & & & & \\
0 & \hat{D}_3 & \hat{E}_3 & \boxed{\hat{F}_3} & \hat{G}_3 & \hat{H}_3 & & & \\
\hat{B}_4 & \hat{C}_4 & \boxed{\hat{D}_4} & \hat{E}_4 & \hat{F}_4 & \boxed{\hat{G}_4} & \hat{H}_4 & & \\
 & \hat{B}_5 & \hat{C}_5 & \hat{D}_5 & \hat{E}_5 & \hat{F}_5 & \hat{G}_5 & \hat{H}_5 & \\
 & & \hat{B}_6 & \boxed{\hat{C}_6} & \hat{D}_6 & \hat{E}_6 & \boxed{\hat{F}_6} & \hat{G}_6 & \hat{H}_6 \\
 & & & \hat{B}_7 & \hat{C}_7 & \boxed{\hat{D}_7} & \hat{E}_7 & \hat{F}_7 & \boxed{\hat{G}_7} \\
 & & & & \hat{B}_8 & \hat{C}_8 & \hat{D}_8 & \hat{E}_8 & \hat{F}_8 \\
 & & & & & \hat{B}_9 & \boxed{\hat{C}_9} & \hat{D}_9 & \hat{E}_9
\end{bmatrix}
$$

Because of the boundary conditions, the elements of $(\overline{\overline{A+B}})$ inside squares will be zero for the 3×3 problem illustrated in figure 13.

The relationships among the elements of $\bar{\bar{L}}$, $\bar{\bar{U}}$, and $(\overline{\overline{A+B}})$ are

$$\alpha = \hat{B} \quad (30a)$$

$$\alpha\delta_{i-1} = \hat{C} \quad (30b)$$

$$\beta = \hat{D} \quad (30c)$$

$$\gamma + \alpha\eta_{i-1} + \beta\delta_{j-1} = \hat{E} \quad (30d)$$

$$\gamma\delta = \hat{F} \quad (30e)$$

$$\beta\eta_{j-1} = \hat{G} \quad (30f)$$

$$\gamma\eta = \hat{H} \quad (30g)$$

where the i and j subscripts refer to the location on the model grid, not in matrix $(\overline{\overline{A+B}})$.

In order to use equations 30a–30g as the basis of a numerical technique for solving equation 24 efficiently by elimination, relationships between the elements of $\overline{\overline{A}}$ and $(\overline{\overline{A+B}})$ must be defined. One possibility is to let the elements correspond exactly and ignore the $\hat{C}$ and $\hat{G}$ diagonal in $(\overline{\overline{A+B}})$. Stone (1968), however, found that this could not be used as the basis of a rapidly convergent iterative procedure. Instead, he defined a family of modified matrices starting with 30b and 30f.

Then the other elements of $(\overline{\overline{A+B}})$ can be defined as equal to the corresponding elements in $\overline{\overline{A}}$ plus a linear combination of $\hat{C}$ and $\hat{G}$. For example

$$\hat{B} = B + \phi_1\hat{C} + \phi_2\hat{G}$$

in which ϕ_1 and ϕ_2 are constants depending on the problem being solved.

What are appropriate linear combinations of $\hat{C}$ and $\hat{G}$ with the elements of $\overline{\overline{A}}$? If equation 27 is written for node (i,j), non-zero coefficients appear not only for the unknowns in the original difference equation but also for $\xi^n_{i-1,j+1}$ and $\xi^n_{i+1,j-1}$. This is illustrated in figure 17. To minimize the effects of the terms introduced in forming the modified matrix equation, $\overline{\overline{B}}\bar{\xi}^n$ for the node (i,j) is defined as

$$\hat{C}[\xi^n_{i-1,j+1} - \omega(\xi^n_{i-1} + \xi^n_{j+1} - \xi^n)] + \hat{G}[\xi^n_{i+1,j-1} - \omega(\xi^n_{j-1} + \xi^n_{i+1} - \xi^n)] \quad (31)$$

where the terms in parentheses are second-order correct approximations for $\xi_{i-1,j+1}$, and $\xi_{i+1,j-1}$, respectively. (See Remson, Hornberger, and Molz, 1971, p. 226, for derivation of these approximations.) To consider these terms good approximations to $\xi_{i-1,j+1}$ and $\xi_{i+1,j-1}$ an iteration parameter, ω, is added. The value of ω ranges between 0 and 1, and its computation is discussed at the end of this section.

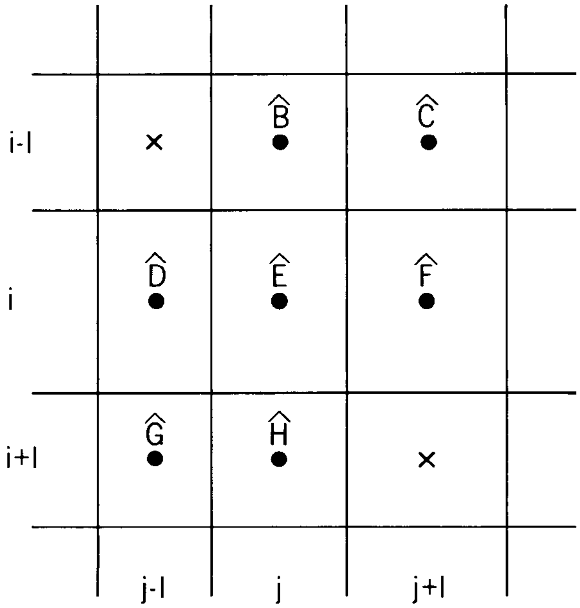


FIGURE 17.—Coefficients of unknowns in equation 27.

With the definition of $\overline{\overline{B}}$ (31), the iteration scheme (equation 27) becomes

$$B\xi^n_{i-1} + D\xi^n_{j-1} + E\xi^n + F\xi^n_{j+1} + H\xi^n_{i+1} + \hat{C}[\xi^n_{i-1,j+1} - \omega(\xi^n_{i-1} + \xi^n_{j+1} - \xi^n)] + \hat{G}[\xi^n_{i+1,j-1} - \omega(\xi^n_{j-1} + \xi^n_{i+1} - \xi^n)] = R^{n-1} \quad (32)$$

Collecting coefficients in equation 32 associated with the nodal positions in the original difference equation gives the desired linear combinations of $\hat{C}$ and $\hat{G}$ with the elements of $\overline{\overline{A}}$ that define the remaining elements of $(\overline{\overline{A+B}})$:

$$\hat{B} = B - \omega\hat{C} \quad (33a)$$

$$\hat{D} = D - \omega\hat{G} \quad (33b)$$

$$\hat{E} = E + \omega\hat{C} + \omega\hat{G} \quad (33c)$$

$$\hat{F} = F - \omega\hat{C} \quad (33d)$$

$$\hat{H} = H - \omega\hat{G} \quad (33e)$$

The coefficient $\hat{C}$ is obtained explicitly by combining equations 33a, 30a, and 30b as

$$\hat{C} = \frac{\delta_{i-1}B}{1 + \omega\delta_{i-1}}. \quad (34a)$$

Finally combining equation 33b and equations 30c and 30f gives

$$\hat{G}=\frac{\eta_{j-1}D}{1+\omega\eta_{j-1}}. \tag{34b}$$

Equations 34, 33 and 30 (in that order) are the first part of the SIP algorithm.

Equation 28 written for node (i,j) is

$$R^{n-1}=Q-(Bh^{n-1}_{i-1}+Dh^{n-1}_{j-1}+Eh^{n-1}+Fh^{n-1}_{j+1}+Hh^{n-1}_{i+1}).$$

As in the Thomas algorithm, the vector $\bar{\xi}^n$ is obtained by a process of forward and backward substitution. Combining equations 27 and 29 gives

$$\overline{\overline{L}}\,\overline{\overline{U}}\bar{\xi}^n=\overline{R}^{n-1} \tag{35}$$

Define an intermediate vector $\overline{V}^n$ by

$$\overline{\overline{U}}\bar{\xi}^n=\overline{V}^n. \tag{36}$$

Then equation 35 becomes

$$\overline{\overline{L}}\,\overline{V}^n=\overline{R}^{n-1}. \tag{37}$$

$\overline{V}^n$ is first computed by forward substitution. This can be seen by writing equation 37 for node (i,j):

$$\alpha V^n_{i-1}+\beta V^n_{j-1}+\gamma V^n=R^{n-1}$$

or

$$V^n=(R^{n-1}-\alpha V^h_{i-1}-\beta V^n_{j-1})/\gamma.$$

The vector $\bar{\xi}^n$ may then be computed by backward substitution. Equation 36 for node (i,j) is

$$\xi^n+\delta\xi^n_{j+1}+\eta\xi^n_{i+1}=V^n$$

or

$$\xi^n=V^n-\delta\xi^n_{j+1}-\eta\xi^n_{i+1}.$$

Stone (1968) recommends an alternating computational procedure. On odd iterations, the equations are ordered in a "normal" manner as shown in figure 13. On even iterations, the numbering scheme is changed to that illustrated in figure 18. This has the effect of making non-zero coefficients appear for the heads $h_{i-1,j-1}$ and $h_{i+1,j+1}$ (the X's in fig. 17) instead of $h_{i-1,j+1}$ and $h_{i+1,j-1}$ and significantly improves the convergence rate. Note that some of the recursion equations are modified by reordering the grid points in the "reverse" manner. The modifications required for the reverse algorithm are

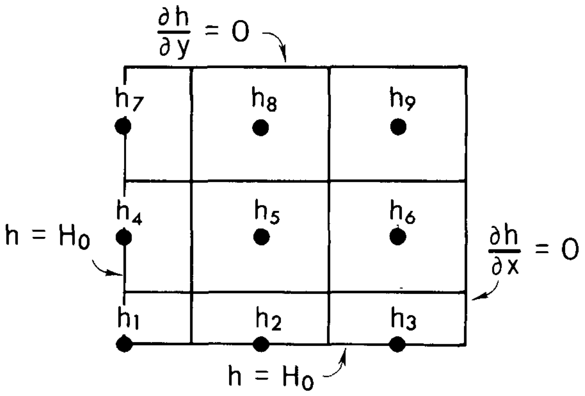


FIGURE 18.—Reverse numbering scheme for 3× 3 problem.

$$\begin{aligned}\hat{C}\ &=\frac{\delta_{i+1}H}{1+\omega\delta_{i+1}};\\ \hat{B}\ &=H-\omega\hat{C};\\ \hat{H}\ &=B-\omega\hat{G};\\ \gamma\ &=E-\alpha\eta_{i+1}-\beta\delta_{j-1};\\ V^n &=(R^{n-1}-\alpha V^n_{i+1}-\beta V^n_{j-1})/\gamma;\\ \xi^n\ &=V^n-\delta\xi^n_{j+1}-\eta\xi^n_{i-1}.\end{aligned}$$

The iteration parameters are computed by equations given in Stone (1968). For variable transmissivity and grid spacing, Stone's equation is

$$(1-\omega_{\max})=\sum_{i=1}^{N_y}\sum_{j=1}^{N_x}\mathrm{Min}\left[\frac{2(\delta x_j)^2}{1+\left(\frac{T_{yy[i,j]}(\delta x_j)^2}{T_{xx[i,j]}(\delta y_i)^2}\right)},\ \frac{2(\delta y_i)^2}{1+\left(\frac{T_{xx[i,j]}(\delta y_i)^2}{T_{yy[i,j]}(\delta x_j)^2}\right)}\right]\div(N_x\times N_y) \tag{38}$$

in which

$\delta x=\Delta x_j/$width of model

$\delta y=\Delta y_i/$length of model

Equation 38 computes an arithmetic average of $\omega_{\max}$ for the algorithm.

The remaining iteration parameters are computed by

$$1-\omega_{l+1}=(1-\omega_{\max})^{l/(L-1)},\ l=0,1,\ldots,L-1$$

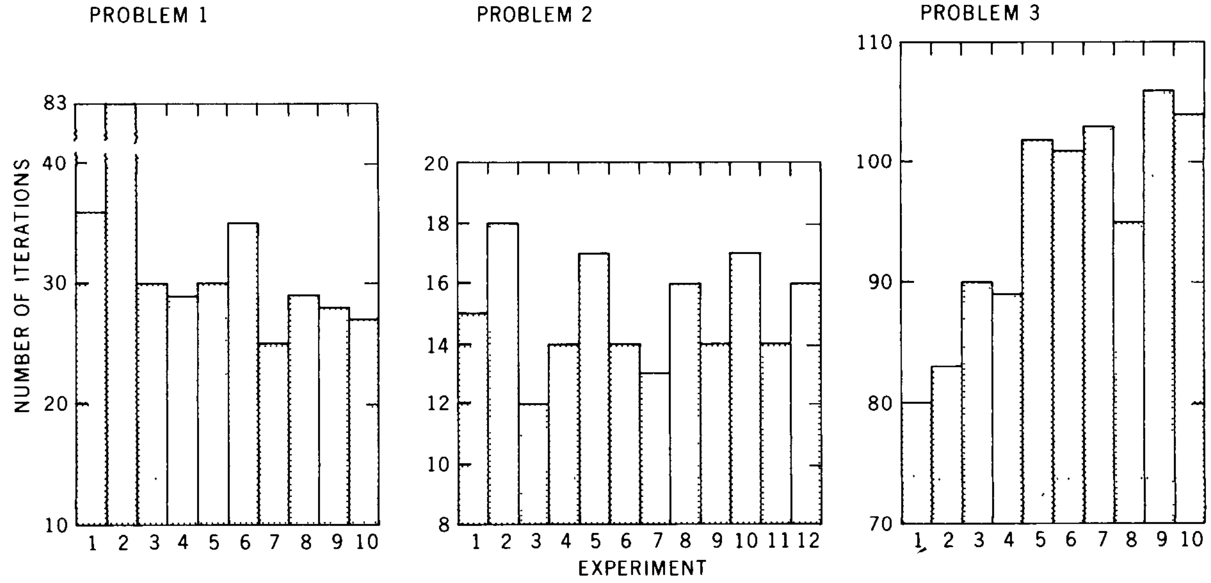
in which L is the number of parameters in a cycle.

Stone (1968) recommends using a minimum of four parameters, each used twice in succession, starting with the largest first. Weinstein, Stone, and Kwan (1969), however, indicate that it is not necessary to start with the largest parameter first or to repeat them.

The results using different numbers and sequences of parameters for the three test problems are shown in figure 19. Except for the sequence 4, 3, 2, 1 in problem 1 the number of iterations required for solution varies up to a maximum of 50 percent for the parameter sequences tested. Several parameter sequences (for example, 1, 2, 3, 4, 5) give convergence near the maximum observed rate for all problems. This result suggests that conducting numerical experiments to determine the best sequence of parameters for a particular problem is generally not justified.

Weinstein, Stone, and Kwan (1969) have a slightly different definition of the maximum parameter ($1-\omega_{max}$=ADI minimum parameter). Their definition of the maximum parameter (which is the maximum over the model, not the arithmetic average of values computed for each node) was used in solving several test problems. In every case convergence was faster using equation 38 to compute the maximum parameter.

Stone (1968) states that a more general form of equation 27 includes another iteration parameter, β', to multiply the term $\bar{R}^{n-1}$. His experience indicated, however, that values of β' other than unity did not generally improve the method. In contrast, the use of β' other than unity has proven to be effective for some of the test problems. In fact, for the fourth problem, a value of β' less than unity is required to obtain a reasonable solution using SIP. Results for problem 4 are not shown in figure 19 because the best sequence of parameters (No. 3) for problem 2 was used in experimenting with the parameter β'.



Experiment	Sequence of parameters
1	1 2 3 4
2	4 3 2 1
3	1 2 3 4 5
4	5 4 3 2 1
5	1 2 3 4 5 6
6	6 5 4 3 2 1
7	1 1 3 3 5 5 2 2 4 4 6 6
8	6 6 4 4 2 2 5 5 3 3 1 1
9	1 2 3 4 5 6 7
10	7 6 5 4 3 2 1
11	1 1 4 4 7 7 2 2 5 5 8 8 3 3 6 6 9 9
12	9 9 6 6 3 3 8 8 5 5 2 2 7 7 4 4 1 1

FIGURE 19.—Iterations required for solution of the test problems by SIP using different numbers and sequences of parameters.

Comparison of Numerical Results

The rate of convergence using different numerical techniques for solving the test problems is compared in figures 20 to 23. The best results from the experiments with each iterative technique are used in the comparisons. Two curves (except for fig. 23) are shown for SIP: one with the parameter $\beta' = 1$ and the other with the best rate of convergence for $\beta' \neq 1$. The sequence of ω parameters is the same for both curves. Two curves are also shown for ADI: one in which the minimum parameter was calculated with equation 23a (indicated by an asterisk in the figures); the other with the best minimum parameter shown on figure 15.

In figures 20 to 23 the absolute value of the maximum residual for each iteration is plotted versus computation time where one unit of work is equal to the time required to complete one SIP iteration. Relative work per iteration is about 1 for ADI, 0.6 for LSOR, and 0.8 for LSOR+2DC. The maximum residual for SIP and ADI fluctuates from a maximum to a minimum over each cycle of parameters. For clarity, the curves connect the local minima for these two methods. Comparisons in figures 20–23 should be made on the basis of the horizontal displacement of the curves, not on the basis of the termination of the curves. This is similar to the type of comparisons made by Stone (1968).

Figure 20 shows the results for problem 1 (10 parameters for ADI, $\omega = 1.87$ for LSOR, $\omega = 1.7$ for LSOR+2DC, parameter sequence, 1,1,3,3,5,5,2,2,4,4,6,6, for SIP). Of the sequence of β' parameters tried, the minimum work required to reduce the residual is obtained with $\beta' = 1.4$, but this is only moderately better than using $\beta' = 1.0$. ADI converges as rapidly as SIP for the first cycles of iteration, but from that point on converges slower than the other iterative techniques. The two ADI curves show about the same rate of convergence for this problem. Next to SIP, LSOR+2DC is most attractive for this problem.

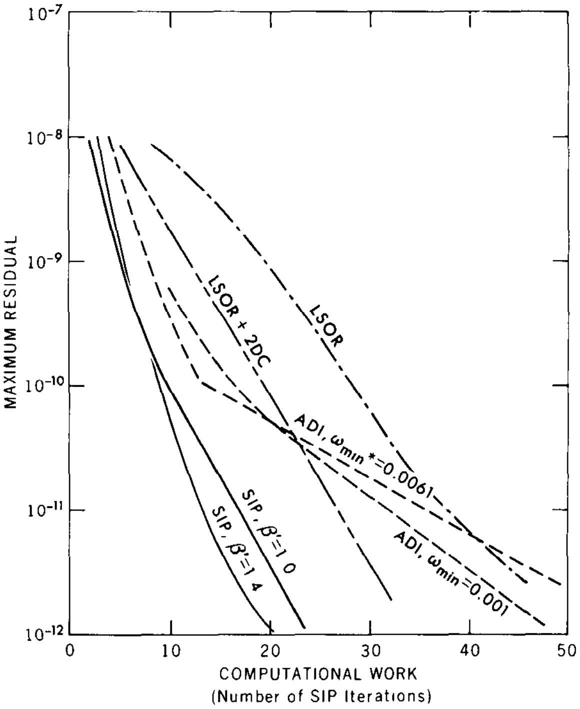


FIGURE 20.—Computational work required by different iterative techniques for problem 1.

The results for problem 2 are shown in figure 21 (10 parameters for ADI, $\omega = 1.6$ for LSOR and LSOR+2DC, parameter sequence 1,2,3,4,5 for SIP). SIP requires the least amount of work for this problem (using $\beta' \neq 1.0$ does not significantly reduce the work required). LSOR and ADI using the best ω_{min} from figure 15 are competitive with SIP. ADI using ω_{min} computed with equation 23a requires about twice as much computational work. LSOR and LSOR+2DC take the same number of LSOR iterations so that the extra work required for 2DC is wasted for this problem.

In figure 22, the results using 4 parameters for ADI, the parameter sequence 1,2,3,4 for SIP, $\omega = 1.88$ for LSOR and $\omega = 1.70$ for LSOR+2DC are plotted for problem 3. In this problem LSOR (with solution lines oriented along columns), ADI with ω_{min} computed with equation 23a, and SIP with $\beta' = 1$ are competitive. Convergence is significantly improved by adding 2DC to LSOR, choosing the best ω_{min} from figure 15 for ADI and letting $\beta' = 1.5$ with SIP.

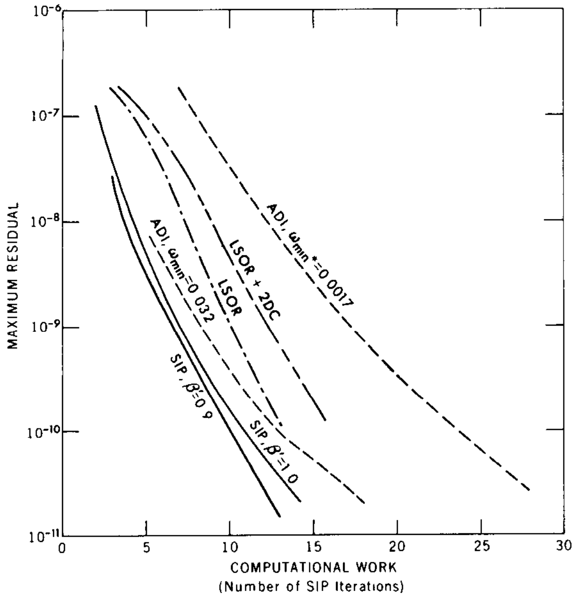


FIGURE 21.—Computational work required by different iterative techniques for problem 2.

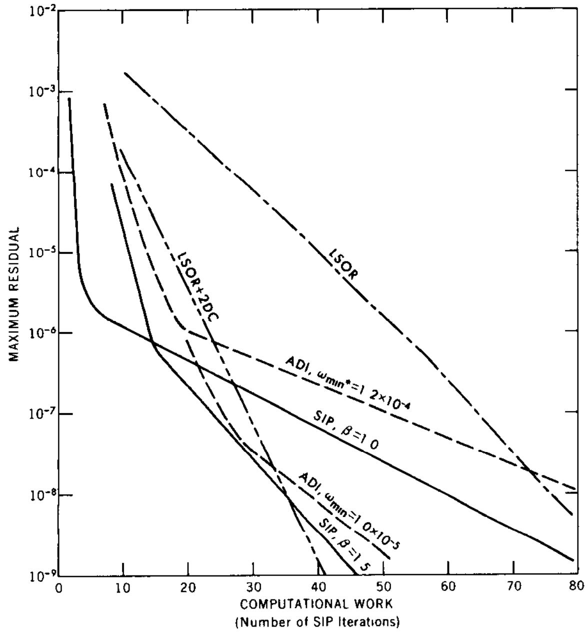


FIGURE 22.—Computational work required by different iterative techniques for problem 3.

The results for problem 4 are shown in figures 23 and 24. The ω iteration parameter sequence for SIP is 1,2,3,4,5, and the two-dimensional correction is applied every fifth iteration for LSOR+2DC. Konikow (oral

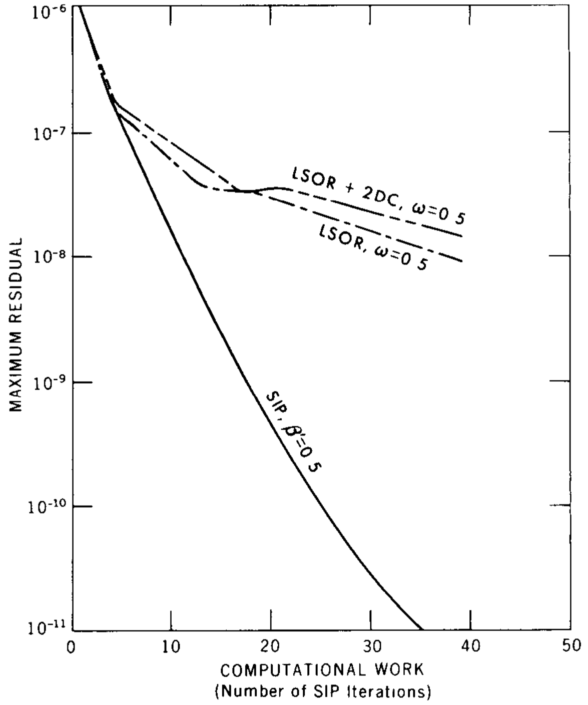


FIGURE 23.—Computational work required by different iterative techniques for problem 4.

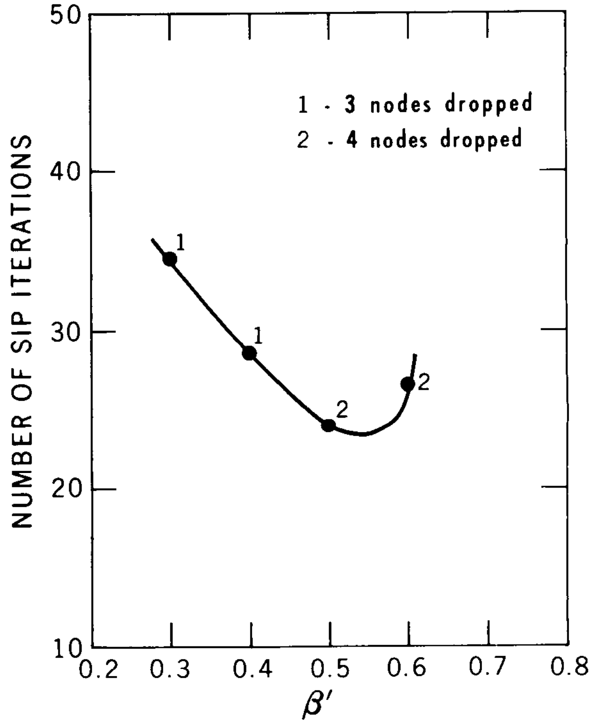


FIGURE 24.—Number of iterations required for solution of problem 4 by SIP using different values of β'.

commun., 1975) was unable to obtain a solution to problem 4 using ADI due to oscillations that eliminated nodes that should have been in the solution. This problem occurred not only with ADI but also with LSOR and LSOR+2DC with $\omega>0.6$ and with SIP with $\beta'>0.6$. The oscillations are apparently caused in part by the nonlinearities of the water-table problem and the necessity to calculate transmissivity at the known iteration level. In a water-table simulation the transmissivity is set to zero and nodes are dropped from the aquifer if the computed head is below the base of the aquifer. For problem 4, at least 3 nodes should be dropped with the initial conditions used.

A solution to problem 4 in which 3 to 4 nodes are dropped is obtained with LSOR and LSOR+2DC when $\omega=0.5$ at the expense of slow convergence. Clearly the most suitable method for this problem is SIP with $\beta'\leqq 0.6$ (fig. 23). In effect the use of $\beta'<1$ for SIP and $\omega<1$ for LSOR represents "underrelaxation" and has the effect of dampening oscillations of head from one iteration to the next. This reduces the tendency for incorrect deletion of nodes from the solution.

Solution of problem 4 emphasizes the advantage of the extra SIP iteration parameter. The optimum value of β' inferred from figure 24 is about 0.5. Note in figure 24 that an additional node is dropped for $\beta'=0.5$ and 0.6. However, the effect of this node on the remainder of the solution is negligible. For $\beta'>0.6$, either convergence was not obtained or excessive numbers of nodes were dropped for those cases that did converge.

The numerical experiments included in this report support the general conclusions of Stone (1968) and Weinstein, Stone, and Kwan (1969) that SIP is a more powerful iterative technique than ADI for most problems. SIP is attractive, not only because of its relatively high convergence rates but because it is generally not necessary to conduct numerical experiments to select a suitable sequence of parameters. SIP has the disadvantage of requiring 3 additional $N_x\times N_y$ arrays.

For the first three problems examined here, ADI is a slightly better technique than LSOR when $\omega_{\min}$ near the optimum is used. Although this result agrees with Bjordammen and Coats (1969) who concluded that ADI is superior to LSOR for the oil reservoir problems they investigated, it is deceptive because less work is required to obtain ω_{opt} for LSOR than is required to find the best $\omega_{\min}$ for ADI by trial and error. Furthermore, LSOR is clearly superior to ADI in application to problem 4 where a solution was not possible with ADI as used in this simulator.

LSOR+2DC seems to be particularly useful with problems dominated by no-flux boundaries. The correction procedure can significantly improve the rate of convergence of LSOR even in problems such as problem 3 where all β_j are zero and non-zero α_i occur for the lower half of the model only.

Considerations in Designing an Aquifer Model

Boundary conditions

An aquifer system is usually larger than the project area. Nevertheless the physical boundaries of the aquifer should be included in the model if it is feasible. Where it is impractical to include one or more physical boundaries (for example, in an alluvial valley that may be several hundred miles long) the finite-difference grid can be expanded and the boundaries located far enough from the project area so that they will have negligible effect in the area of interest during the simulation period. The influence of an artificial boundary can be checked by comparing the results of two simulation runs using different artificial 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. This indicates to the program that these nodes are to be skipped in the computations.

A constant flux may be zero (impermeable boundaries) or have a finite value. A zero-flux boundary is treated by assigning a value of 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 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 (or discharge) wells to the appropriate nodes. Figure 25 illustrates various types of boundary conditions.

The type of boundaries appropriate to the field problem may require careful consideration. In particular, should streams be treated as constant-head boundaries or are they more realistically treated as partially penetrating with a leaky streambed? If a leaky streambed is used, note that the leakage occurs over the area of the blocks assigned to the stream. If the area of the streambed is less than the area of the blocks, the ratio of streambed hydraulic conductivity to thickness can be proportionately reduced to make the amount of leakage realistic.

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 wells. For this objective in a confined aquifer for which the equations are linear, there is no need to impose the natural flow system as the initial condition since the computed drawdown can be superimposed on the natural flow system, if desired.

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 water levels will change during the simulation, not only in response to the new pumping stress, but also due to the initial conditions. This may or may not be the intent of the user.

To start from steady-state conditions in which flow is occurring, the model can be used to compute the initial head by leaving out the new stress (for example, wells) and setting all storage terms to zero. This is also a useful calibration procedure to compute unknown terms such as the ratio of hydraulic conductivity to thickness for leakage.

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 respective 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 area of the cell.
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 X direction, experience has shown that restricting the ratio $\Delta X_j/\Delta X_{j-1} \leqq 1.5$ will avoid large truncation errors and possible convergence problems.
3. Nodes should be placed close together in areas where there are spatial changes in transmissivity. For example, in cross-sectional problems with aquifers separated by confining beds, many layers of nodes are required in the confining bed to obtain a good approximation of the head distribution (and consequently the flux) during transient conditions.
4. The grid should be oriented so that a minimum of nodes are outside the aquifer. The orientation of the grid with respect to latitude and longitude or some other geographic grid system would be a secondary consideration. However, if the aquifer is anisotropic, the grid should be oriented with its axes parallel to the principal directions of the transmissivity tensor. Otherwise,

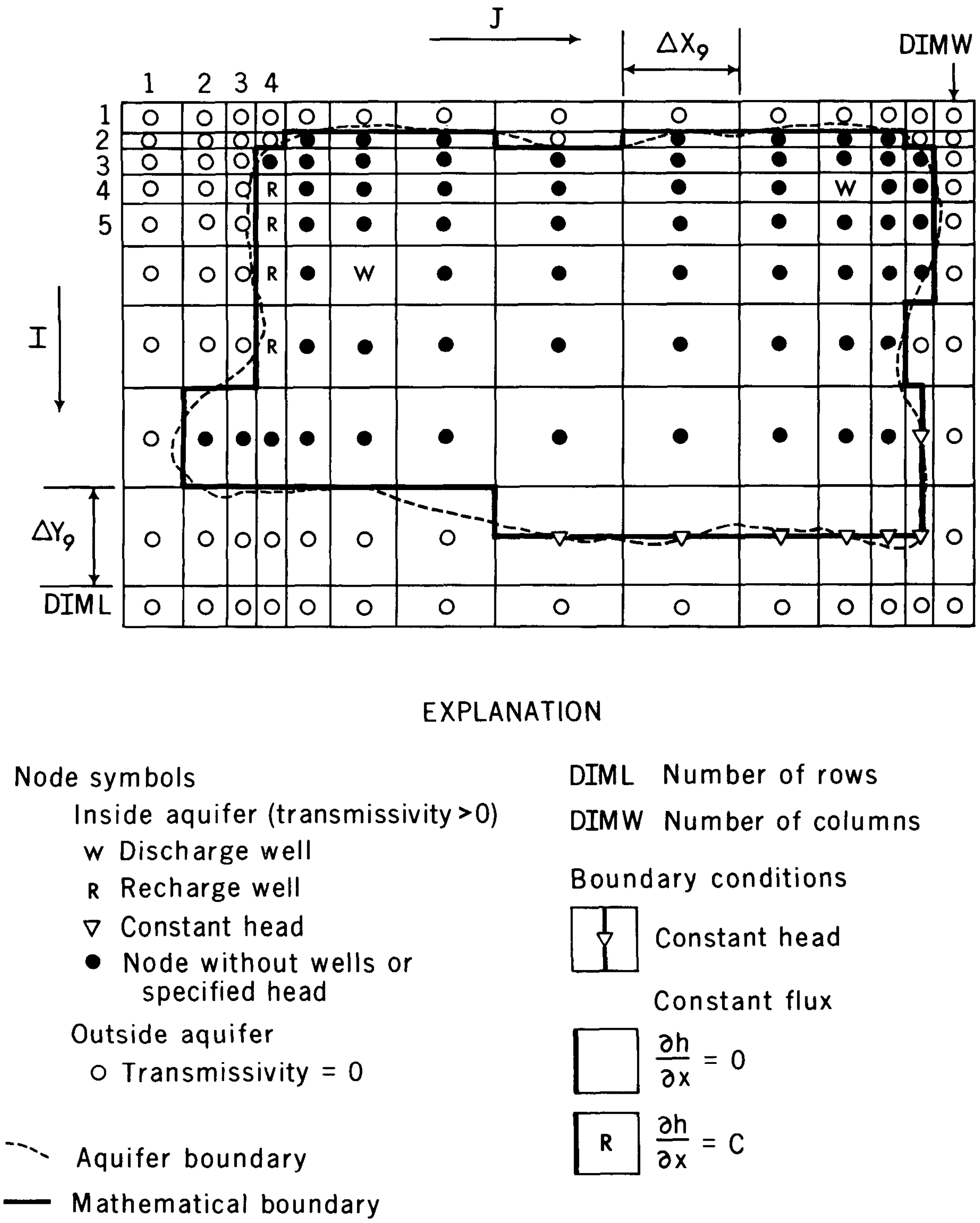


FIGURE 25.—Variable, block-centered grid with mixed boundary conditions.

the flow equation would include cross-product terms and the solution would be restricted to ADI and LSOR because additional diagonals appear in the coefficient matrix and SIP, in its usual form, cannot be used.

5. The rows should be numbered in the short dimension for the alphameric plot on the line printer or for plotting data with an *X–Y* 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 fig. 26.)
6. The core requirements and computation time are proportional to the number of nodes representing the aquifer.

Selected References

Akbar, A. M., and Arnold, M. D., and Harvey, O. H., 1974, Numerical simulation of individual wells in a field simulation model: Soc. Petrol. Eng. Jour., Aug. 1974, p. 315–320.

Aziz, K., and Settari, A., 1972, A new iterative method for solving reservoir simulation equations: Jour. Canadian Petrol. Technology, Jan.–Mar. 1972, p. 62–68.

Bjordamman, J., and Coats, K. H., 1969, Comparison of alternating-direction and successive over-relaxation techniques in simulation of reservoir fluid flow: Soc. Petrol. Eng. Jour., March, 1969, p. 47–58.

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

Breitenbach, E. A., Thurnau, D. H., and van Poollen, H. K., 1969, Solution of the immiscible fluid flow simulation equations: Soc. Petrol. Eng. Jour., June 1969, p. 155–169.

Carnahan, B., Luther, H. A., and Wilkes, J. O., 1969, Applied numerical methods: New York, John Wiley and Sons, 604 p.

Carré, B. A., 1961, The determination of the optimum accelerating factor for successive over-relaxation: Comp. Jour., v. 4, p. 73–78.

Cooley, R. L., 1974, Finite element solutions for the equations of groundwater flow: Desert Research Institute Univ. Nevada, Tech. Report Series, H–W, Hydrology and Water Resources Pub. No. 18, 134 p.

Cosner, O. J., and Horwich, E., 1974, Grid-coordinate generation program: U.S. Geol. Survey open-file report, 27 p.

Freeze, R. A., 1971, Three-dimensional, transient, saturated-unsaturated flow in a groundwater basin: Water Resources Research, v. 7, no. 2, p. 347–366.

Hantush, M. S., 1960, Modification of the theory of leaky aquifers: Jour. Geophys. Research, v. 65, no. 11, p. 3713–3725.

Konikow, L. F., 1974, Modeling mass transport in a shallow aquifer: Am. Geophys. Union Trans., v. 55, no. 4, p. 256.

——— 1975, Hydrogeologic maps of the alluvial aquifer in and adjacent to the Rocky Mountain Arsenal, Colorado: U.S. Geol. Survey open-file report, 74–342.

Peaceman, D. W., and Rachford, H. H., Jr., 1955, The numerical solution of parabolic and elliptic differential equations: Soc. Indust. Appl. Math. Jour., v. 3, no. 1, p. 28–41.

Pinder, G. F., 1969, An iterative digital model for aquifer evaluation: U.S. Geol. Survey open-file report, 43 p.

——— 1970, A digital model for aquifer evaluation: U.S. Geol. Survey Tech. Water-Resources Investigations, Book 7, Chap. C1, 18 p.

Pinder, G. F., and Bredehoeft, J. D., 1968, Application of a digital computer for aquifer evaluation: Water Resources Research, v. 4, no. 5, p. 1069–1093.

Prickett, T. A., 1967, Designing pumped well characteristics into electric analog models: Ground Water, v. 5, no. 4, p. 38–46.

Prickett, T. A., and Lonnquist, C. G., 1971, Selected digital computer techniques for groundwater resources evaluation: Illinois State Water Survey Bull. 55, 62 p., Urbana.

Remson, I., Hornberger, G. M., and Molz, F. J., 1971, Numerical methods in subsurface hydrology: New York: Wiley-Interscience, 389 p.

Stone, H. K., 1968, Iterative solution of implicit approximations of multidimensional partial differential equations: Soc. Indust. Appl. Math., Jour. Numer. Anal., v. 5, no. 3, p. 530–558.

Thiem, G., 1906, Hydrologische Methoden: Leipzig, Gebhardt, 56 p.

Trescott, P. C., 1973, Iterative digital model for aquifer evaluation: U.S. Geol. Survey open-file report, 18 p.

Varga, R. S., 1962, Matrix iterative analysis: New Jersey, Prentice-Hall, 322 p.

von Rosenberg, D. U., 1969. Methods for the numerical solution of partial differential equations: New York, Elsevier, 128 p.

Watts, J. W., 1971, An iterative matrix solution method suitable for anisotropic problems: Soc. Petrol. Eng. Jour., March 1971, p. 47–51.

——— 1973, A method for improving line successive overrelaxation in anisotropic problems—a theoretical analysis: Soc. Petrol. Eng. Jour., April 1973, p. 105–118.

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

COMPUTER PROGRAM AND RELATED DATA

Attachment I

Notation

$\overline{\overline{A}}$	coefficient matrix;
$\overline{\overline{A}}_c, \overline{\overline{A}}_r$	coefficient matrices for ADI column and row equations;
$\overline{\overline{A}}_\lambda$	LSOR coefficient matrix;
b	saturated thickness of the aquifer (L);
$\overline{\overline{B}}$	modifying matrix for SIP;
B, D, E, F, H	coefficients in difference equation;
$\hat{B}, \hat{C}, \hat{D}, \hat{E}, \hat{F}, \hat{G}, \hat{H}$	coefficients of $(\overline{\overline{A+B}})$;
B', D', E', F', H'	coefficients of equations defining 2DC;
BE, G, W	notation used in Thomas algorithm;
E_r, E_c	coefficients in equations defining ADI;
ET_z	depth below land surface at which evapotranspiration ceases (L);
G	elevation of the land surface (L);
h	hydraulic head (L);
$h^{\dagger}$	intermediate head value (L);
h^{n*}	corrected head at iteration n (L);
$h_{i,j,o}$	initial head in the aquifer (L);
$\hat{h}_{i,j,o}$	hydraulic head on the other side of the confining bed (L);
h_w	hydraulic head in a well (L);
$H_{i,j}$	saturated thickness of the aquifer at radius $r_e (L)$;
H_w	saturated thickness of the aquifer at radius $r_w (L)$;
i	index in the y dimension;
j	index in the x direction;
k	time index;
K_{xx}, K_{yy}	principal components of the hydraulic conductivity tensor (Lt^{-1});
K'	hydraulic conductivity of the confining bed (L/t);
l	iteration parameter index;
L	number of iteration parameters in a cycle;
$\overline{\overline{L}}$	lower triangular factor of $(\overline{\overline{A+B}})$;
m	thickness of the confining bed (L);
M	vector of ADI parameters;
n	iteration index;
N_s	number of arrays required for the options;
N_x	number of nodes in a row;
N_y	number of nodes in a column;
q'	flux from a confining bed (Lt^{-1});
$q^{\cdot}$	transient part of q' (Lt^{-1});
q_{et}	evapotranspiration flux (Lt^{-1});
q_{re}	recharge flux (Lt^{-1});
Q	known term in difference equation;
Q_{et}	maximum evapotranspiration rate (Lt^{-1});
Q_c, Q_r	known terms in equations defining ADI;
Q_w	well discharge (L^3t^{-1});
Q_λ	known term in equation defining LSOR;
r	radial distance from center of pumping well (L);
r_e	effective radius for a well block (L);
r_w	well radius (L);
r_1	radius equivalent to the average grid spacing for the well block (L);
R	residual;
R^{n*}	residual computed for 2DC;
R'	sum of residuals for 2DC;
R_c, R_r	residuals for ADI column and row computations;
R_λ	LSOR residual;
S	storage coefficient (dimensionless);
S_s	specific storage of the confining bed (L^{-1});
S_y	specific yield (dimensionless);
t	elapsed time of the pumping period (t);
T	transmissivity (L^2t^{-1});
T_L	transient leakage coefficient (t^{-1});
$T_{xx}, T_{xy}, T_{yx}, T_{yy}$	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;
$W(x,y,t)$	volume flux per unit area (Lt^{-1});
α	row correction for LSOR;
$\alpha, \beta, \gamma, \delta, \eta$	elements of factors of $(\overline{\overline{A+B}})$;
β	parameter in Hantush (1960) solution;
$\hat{\beta}$	column correction for LSOR;
β'	iteration parameter for SIP;
γ	constant used in calculating ADI parameters;
δ_x, δ_y	normalized grid spacing;
Δh	head change between adjacent nodes (L);
Δt	time increment (t);
Δx	space increment in the x direction (L);
Δy	space increment in the y direction (L);
ε	closure criterion (L);

$\overline{\zeta}$	vector of change in head over an iteration;
$\rho(G)$	spectral radius of Gauss-Seidel iteration matrix;
ϕ_1, ϕ_2	constants in definition of coefficients of $(\overline{\overline{A+B}})$;
ω	acceleration parameter;
ω_l	iteration parameter;
ω_{max}	maximum iteration parameter;
ω_{min}	minimum iteration parameter;
ω_{opt}	optimum acceleration parameter.

Attachment II, Computer Program

Main program

The first function of the main program is to dimension the arrays for the field problem being simulated. The algorithm allocates storage space reserved in a vector, Y. Some arrays are required for every simulation; others are needed only if certain options are specified. The information needed to allocate space to the arrays is contained in the Group I data cards which are read by the main program (see Attachment III).

Once the model is compiled, it does not need to be recompiled for a new field problem unless (1) the logic is changed or (2) the vector Y is not dimensioned large enough for the new problem. The minimum dimension of the vector Y (YDIM) can be computed by

$$\text{YDIM} \cong (15+N_a)N_xN_y \qquad (39)$$

in which N_a is the total number of arrays required for the options (from table 2).

Equation 39 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'.

Table 2.—Number of arrays required for the options

Option	Number of arrays
Water Table	3
Conversion[1]	1
Leakage	3
Evapotranspiration	1
SIP	4

[1] Conversion also requires the arrays for the water table option.

In the second part of the main program, the location of the initial addresses of the arrays are passed to the subroutines. (See table 3 for details.) The variables in table 3 defining the dimensions of the arrays are defined in Attachment VI; the first four arrays and XII are double precision.

The last part of the main program controls the sequence of computations illustrated by the generalized flow chart (Appendix V). In the flow chart, the routines are lettered in sequence starting with the main program. Entry points for the routines are numbered in sequence along the left side of the chart. Exits from a routine are indicated by circles containing the entry point of the routine to which control passes. A break occurs in the flow chart following an unconditional exit. Variables used in the flow chart are defined in Attachment VI.

Subroutine DATAI

Instructions for the preparation of the data deck are given in Attachment 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 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 (for example, recharge rate) with this set of cards.

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 decline decreases during a pumping period, the time step is increased by the factor CDLT each step (commonly 1.5). For

Table 3.—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	×	×	×	×	×	×	(IZ, JZ) *8
BE	2	--	--	×	--	--	--	IMAX *8
G	3	--	--	×	--	--	--	IMAX *8
TEMP	4	--	--	×	--	--	--	IMAX *8
KEEP	5	--	×	×	×	×	--	IZ, JZ
PHE	6	--	--	×	×	×	--	IZ, JZ
STRT	7	×	×	×	×	×	--	IZ, JZ
SURI	8	×	×	--	×	--	×	IZ, JZ
T	9	×	×	×	×	×	--	IZ, JZ
TR	10	×	--	--	×	×	--	IZ, JZ
TC	11	×	--	--	×	×	--	IZ, JZ
S	12	×	--	×	×	×	×	IZ, JZ
QRE	13	×	--	×	--	×	--	IZ, JZ
WELL	14	×	×	×	×	×	×	IZ, JZ
TL	15	×	--	×	×	×	--	IZ, JZ
SL	16	×	--	×	×	--	--	IZ, JZ
PERM	17	×	×	--	×	×	--	IP, JP
BOTTOM	18	×	×	--	×	×	--	IP, JP
SY	19	×	--	--	×	×	--	IP, JP
RATE	20	×	--	--	×	×	--	IR, JR
RIVER	21	×	--	--	×	×	--	IR, JR
M	22	×	--	--	×	×	--	IR, JR
TOP	23	×	×	--	×	×	--	IC, JC
GRND	24	×	--	--	×	×	--	IL, JL
DEL	25	--	--	×	--	--	--	IS, JS
ETA	26	--	--	×	--	--	--	IS, JS
V	27	--	--	×	--	--	--	IS, JS
XI	28	--	--	×	--	--	--	IS, JS
DELX	29	×	×	×	×	×	×	JZ
DDN	30	--	×	--	--	--	--	JZ
BETA	31	--	--	×	--	--	--	JZ
DELY	32	×	×	×	×	×	×	IZ
ALFA	33	--	--	×	--	--	--	IZ
WR	34	×	×	--	--	--	--	IH
NWR	35	×	×	--	--	--	--	IH, 2
XII	36	--	--	×	--	--	--	IMAX *8
TEST 3	37	--	×	×	--	--	--	IMX1

any time step (k) the time increment is given by

$$DELT_k = CDLT * DELT_{k-1}.$$

$DELT_0$ 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 $DELT_0$, and set NUMT greater than the expected number of time steps. The program computes the required initial $DELT_0$ (which will not exceed the value of $DELT_0$ coded on card 1 of group IV) and NUMT to arrive exactly at TMAX on the final time step. In a simulation of one pumping period in which results are required at several specific times, the simulation can be broken into several "pumping periods." Each period will have the same pumpage, and TMAX is used to specify the appropriate times for display of results.
2. To simulate a given number of the time steps, set TMAX greater than the expected simulation period and the program will use $DELT_0$, 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 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 multistep simulation so that the results can be compared.

For steady-state simulations, set the storage coefficient and (or) specific yield of the aquifer and the specific storage of the confining bed 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. If the calculations do not converge to a solution within a reasonable number of iterations, it may be necessary to use a transient simulation for enough steps to attain steady state (see also the discussion of ADI iteration parameters) or use another numerical technique.

Initialization

In addition to reading data and computing the time parameter, this routine initializes other arrays and scalar parameters. In particular, note that the leakage coefficient, *TL*, will equal $K'_{i,j}/m_{i,j}$ and can be computed once for the entire simulation if the specific storage of the confining bed is zero. The computation of the steady leakage term, *SL*, and the division of well discharge by the area of the cell need to be done only once for each pumping period. At the beginning of each pumping period the starting head (STRT) and the simulation time (SUMP) used in computing transient leakage are initialized.

Subroutine STEP

Subroutine STEP initializes variables for a new time step, checks for steady-state conditions after a solution is obtained for the time step, and controls the printing and punching of results and the writing of results on disk. If head values are punched at the end of the simulation or are written on disk, they can be used to extend the simulation or as input to plotting routines. (See the program by Cosner and Horwich, 1974.) Currently, a general program is being written to display results in various forms on the line printer and plotters; it is described in detail in another section of this report.

In the check for steady state during transient simulations, the head change over a time step is computed. If the absolute value of change at all nodes is less than EROR, the message 'STEADY STATE AT TIME STEP X' is printed. The program then prints all desired output for the final time step (X) and proceeds to read data for the next pumping period, if any.

Maximum head change for each iteration

The printed results are explained in the section on theory and in the discussions of subroutines COEF, CHECKI, and PRNTAI or are self explanatory, except for the listing of the absolute value of the maximum head change for each iteration. This information is useful if convergence is slow with ADI or SIP because it may indicate that a slightly larger error criterion will give a satisfactory solution with considerably fewer iterations.

Subroutine SOLVE

The three SOLVE routines, SOLVE1, SOLVE2, and SOLVE3 are, respectively, SIP, LSOR and ADI. They have been described in previous sections, but a few additional comments are necessary.

In these routines and in subroutine COEF, the usual (I,J) 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 a double subscript and, as a consequence, computational efficiency is improved. The five variables used as subscripts in this notation are defined in Attachment VI.

SIP iteration parameters

The algorithm in ITER1 permits computation of the iteration parameters in increasing or decreasing order and repeat of parameters depending on the initialization of the vector IORDER. Note that LENGTH is twice the number of different parameters and that the DATA statement that initializes IORDER assumes LENGTH = 10. Replace the DATA statement with a READ statement if additional flexibility is desired in choosing the order of parameters without recompiling the subroutine.

Exceeding permitted iterations

If the permitted number of iterations for a time step is exceeded, the message 'EXCEEDED PERMITTED NUMBER OF ITERATIONS' is printed. Following the message the mass balance, head matrix, etc., as specified in the options are printed for the final iteration. This information is useful in determining the cause of the nonconvergence. Before terminating the run, the mass balance and head values will be punched if PUNC was specified in the options or written on disk if IDK2 was specified. With punched output or results on disk, the user has the option to extend the number of iterations if it appears that a solution can be obtained. If iterations are exceeded on the first time step, the head values saved (punched or written on disk) were computed in the last iteration. If iterations are exceeded on a subsequent time step, KT, the head values and mass-balance parameters saved are the results for time step KT–1.

Subroutine COEF

Most of the calculations for coefficients used in the solution of the numerical schemes are done in this routine. The more extensive computations except those described in the section on theory are discussed in the following paragraphs.

Transient leakage coefficients

The algorithm for the transient parts of equations 9 and 10 is the same except for two conditional statements that recompute PPT and DENOM if dimensionless time is in the range for applying equation 9. In performing the infinite summation, the code checks for the significance of additional terms, but in any case limits the summation to a maximum of 200 terms. The minimum and maximum values of dimensionless time, TMIN and TT, are retained and printed with the results for the time step so that the user will know whether or not transient leakage effects are significant.

Transmissivity as a function of head

The transmissivity for water-table or combined water-table–artesian aquifers is computed as a function of the saturated thickness of the aquifer. If a cell (except a cell with well discharge) goes dry, a message 'NODE I, J GOES DRY' is printed, the transmissivity for the cell is set to zero, and the head is set to the initial surface (so that the location of the cell will show up in the output). No provision is made to permit the cell to resaturate in subsequent pumping periods because the additional code necessary to accommodate this special situation is not warranted in a general program.

When a cell with well discharge goes dry (that is, a hypothetical well with radius r_e goes dry), the program terminates the computation with printed output, and, if specified in the options, saves the results. Printed output is headed by 'WELL I, J GOES DRY' followed by drawdown when the well went dry. If results for the previous time step were not printed, drawdown and a mass balance (if specified in the program options) for the previous time step are printed. Finally, if specified in the options, mass-balance parameters and head values for the previous time step are punched or written on disk so that the user has the option of continuing the simulation after modifying the well discharge.

TR and TC coefficients

The TR and TC arrays save values that are used repeatedly in the algorithm. They are computed once for artesian problems and each iteration for water-table and combined artesian–water-table simulations. TR (I,J) is the harmonic mean of T_{xx}(I,J)/DELX(J), T_{xx}(I,J+1)/DELX(J+1); TC (I,J) is the harmonic mean of T_{yy}(I,J)/DELY(I), T_{xx} (I+1,J)/DELY(I+1).

Subroutine CHECKI

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.

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 aquifer. Other computations in the algorithm are self explanatory.

The difference between the sum of sources and sum of discharges from the system is usually less than 1 percent. A larger error, however, does not necessarily mean that the results are poor; it may be due to lack of precision in calculating the mass balance. This has been observed, for example, if a leaky streambed is given a large K'/m ratio so that it is effectively a constant-head boundary. The leakage computation is inaccurate if the head values at a stream node are identical to 6 or 7 significant figures and they are stored as single precision variables.

To the right of the cumulative mass balance are printed the flow rates for the current time step. They are self explanatory except for leakage. "Leakage from previous pumping period" is the leakage resulting from gradients across the confining bed at the start of the current pumping period. The "total" leakage is the sum of leakage due to the initial gradients plus leakage induced by head changes during the current pumping period.

Subroutine PRNTAI

This routine prints a map of drawdown and hydraulic head. Up to three characters are plotted for each cell with the rightmost character as close to the location of the node as the printer will allow. An option to permit the printing of results at different scales in the x and y dimensions is useful for cross sections. This routine is useful for displaying results during calibration runs. More elegant graphical displays for final results are described in another section.

The user specifies XSCALE and YSCALE, the multiplication factors required to change from units used in the model to units used on the map; 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; and MESUR, the name of the unit used on the map. 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 at 1 foot increments and head values at 10 foot increments are to be plotted. Then XSCALE=YSCALE=5280, DINCH=3, FACT1=1, FACT2=0.1; and MESUR=MILES.

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 fig. 26.) 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 cell boundaries depending on the scaling. The map is automatically centered on the page and is limited to a maximum of 12 inches (300 mm) in the Y direction. If the parameters for a map are specified such that the Y dimension is more than 12 inches (300 mm) 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 suit-

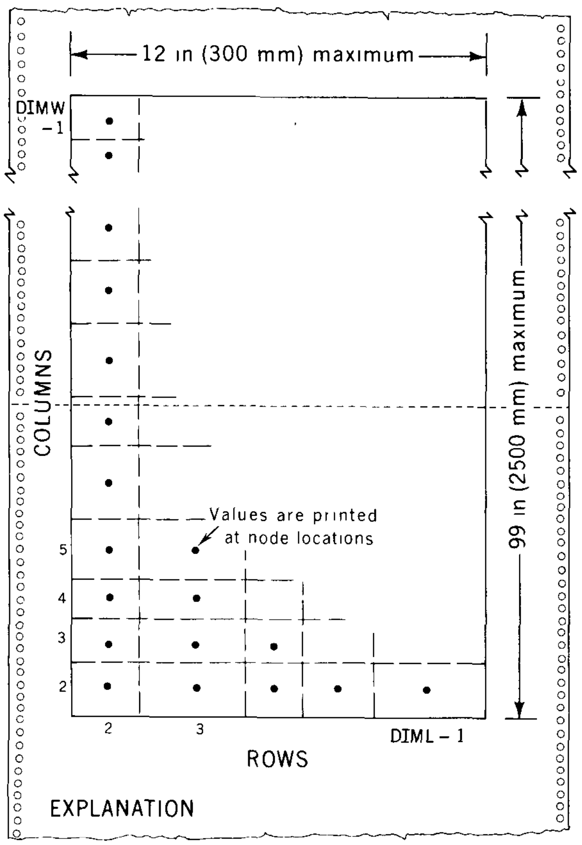


FIGURE 26.—Orientation of map on computer page.

able 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 (2500 mm).) Several parameters (PRNT, BLANK, N1, N2, N3, and XN1) are initialized in the BLOCK DATA routine to values that assume the line printer 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 routine

The BLOCK DATA routine initializes scalar parameters and arrays used in PRNTAI and other subroutines. The unit numbers for card reader, line printer, and card punch are commonly 5, 6 and 7, respectively. At computer installations where other numbers are used, change the initialization of P, R, and PU.

Technical information

Storage requirements

Using the FORTRAN G, Level 21 compiler, the source code and fixed-dimension arrays require 100K bytes of memory (88K bytes if only one SOLVE routine is complied). The storage requirements including all options but not including storage requirements for reading and writing on disk are (100+X/256) K bytes where X is the dimension of the vector Y in the main program. Subtract 14K bytes from the values if the FORTRAN H, OPT=2 compiler is used. The FORTRAN G compile step requires 120K bytes of memory and the FORTRAN H, OPT=2 compiler requires 218K bytes of memory.

Computation time

Computation time is a function of so many variables that no general rule can be stated. For example, the simulation of a nonlinear water-table problem requires many more computations per time step than does the simulation of a linear artesian-aquifer problem.

As an example, the simulation of a linear aquifer system (problem 2) with a grid of 25×38 required 45 seconds for 40 iterations with the program compiled under FORTRAN G. This is about 0.002 seconds for each node inside the aquifer each iteration on the IBM 370/155. A significant reduction (about 1/3) in execution time can be achieved by using the FORTRAN H compiler which generates a more efficient code than the FORTRAN G compiler.

Further significant reductions in execution time can be achieved if the model is designed for a specific problem. Problem 3, for example, does not require computation of leakage, storage, or evapotranspiration terms.

Use of disk facilities for storage of array data and interim results

In an effort to expedite use of the program on remote terminals connected to the IBM 370/155, options are included to utilize disk storage facilities. These options enable storage and retrieval of array data (STRT, PERM, and so forth) 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 MAN0480) 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(14,???,U,KKK)
                                  MAN0480
```

where ??? is the number of nodes for the problem being solved (DIMLxDIMW). 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 provided by the system (IBM 370/155) for semipermanent 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.

```
//GØ. FT02F001 DD DSN=Azzzzzz.AZbbb.
         cxxwwwww.aaaaaaaa,
//     UNIT=ØNLINE,DISP=(NEW,
         KEEP),
//     SPACE=(????,(14)),DCB=
         (RECFM=F)
```

where

zzzzzz	are the first six digits of a nine digit account number;
bbb	are the last three digits of a nine digit account number;
c	is the center code (same as column 3 on job card);
xx	is the two digit organization code (same as columns 4 and 5 on job card);
wwwww	is the four or five digit program number (same as the program number beginning in column 24 of the job card);
aaaaaaaa	is any 1 to 8 character name used to designate the name of the data set;
????	is the number of bytes per record that are to be reserved and should be set equal to DIMLxDIMWx4.

The instructions for the DSN parameter are also given in the CCD users manual, chapter 5, pages 3 and 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:

```
//GØ. FT02F001 DD DSN=Azzzzzz.Azbbb.
         cxxwwwww.aaaaaaaa,
```

```
//     UNIT=ØNLINE, DISP=SHR,VØL
          =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 will be discussed completely in the section on Data Deck Instructions (Attachment III).

If use of this option is selected, space for buffers must be reserved via the REGION parameter on the EXEC card. The amount of space needed is approximately equal to two times the number of bytes per record (indicated in the SPACE parameter on the DD card defined above).

Interim results (head values, cumulative simulation time, and mass-balance parameters) can be punched on cards or can be stored and retrieved from data sets on disk in much the same manner as array data. Use of storage on disk is initiated by parameters on the simulation options card. (See attachment III, card 3.)

Definition of the sequential data set on disk where the information will be stored is accomplished by a DD statement in the JCL string used to execute the program. If one of the system disk packs is used to store the data set, the first reference to the data will be different from subsequent references as in the case of array data sets. The first reference will take the following form if the FORTGCG procedure is used.

```
//GØ.FT04F001 DD DSN=Azzzzzz.AZbbb.
          cxxwwwww.aaaaaaaa,
//     UNIT=ØNLINE, DISP=(NEW,
          KEEP),SPACE=(TRK,(1,1),
          RLSE),
//     DCB=(RECFM=VBS,LRECL
          =dddd, BLKSIZE=eeee)
```

The DSN parameter is defined in the same manner as previously discussed for the direct access (array) data sets and:

dddd—equals DIMLxDIMWx8+48 ($\leqq$6440)
eeee—equals DIMLxDIMWx8+52 ($\leqq$6440)

If BLKSIZE (eeee) exceeds 6444, code 6444 for (eeee) and 6440 for (dddd). Also, additional core equal to about two times the value of BLKSIZE must be reserved for buffers via the REGION parameter on the EXEC card.

Once the initial reference to the data set has been successfully processed, the system will indicate (via the JCL printout) on what volume the data set has been established (for example, SYS011 or SYS015) and, subsequent references to the data set will appear as follows:

```
//GØ. FT4F001 DD DSN=Azzzzzz.AZbbb.
          cxxwwwww.aaaaaaaa,
//     UNIT=ØNLINE, VØL=SER=
          yyyyyy,DISP=SHR
```

where yyyyyy is the name of the disk pack (for example, SYS011) that contains the data set and DSN is as previously described.

To destroy (erase) an array data set or an interim results data set, simply execute the following job.

```
// EXEC PGM=IEFBR14
//X DD DSN=Azzzzzz.AZbbb.cxxwwwww.
     aaaaaaaa,
// UNIT=ØNLINE, VØL=SER=yyyyyy,
     DISP=(ØLD,DELETE)
```

Use of the disk facilities is illustrated in Appendix IV.

Graphical display package

A series of computer programs are currently being written and assembled that will enable graphical display of results of computer models. Components of this graphical display package will include:

1. time-series plots of model results on the printer,
2. time-series plots on pen plotters (CALCOMP),
3. contour maps of model results at selected time steps on the printer,
4. contour maps utilizing pen plotters, and
5. other graphical displays, such as perspective (three-dimensional) drawings.

The FORTRAN code shown in figure 27 can be inserted into the program to produce output that can be used in the graphical display package. The changes to MAIN and STEP are required after statements MAN2600 and STP1000, respectively. Statement MAN2600 is deleted. In subroutine PRNTAI, the REAL*8 specification and the DIMENSION statement must be added and the remaining code inserted after statement PRN1650. Also, unit numbers 10 and 11 must be specified on DD statements when the program is executed. Unit 10 is used only for temporary storage and the following DD statement will generally suffice.

```
//GØ.FT10F001 DD DSN=&&DATA,DISP
             =(NEW,DELETE),UNIT
             =ØNLINE,
//    SPACE=(TRK,(10,5)),DCB=
             (RECFM=VBS,LRECL=6440,
             BLKSIZE=6444)
```

Unit 11 points to the data set that is used to store the data required by the graphical display package and must be semipermanent in nature. That is, it must not be deleted upon completion of your job. The DD statement will generally take the following form.

```
//GØ.FT11F001 DD DSN=Azzzzzz.AZbbb.
             cxwwwww.aaaaaaaa,
//    DISP=(NEW,KEEP),UNIT=
             ØNLINE,SPACE=(TRK,(10,5),
             RLSE),
//    DCB=(RECFM=VBS,LRECL=6440,
             BLKSIZE=6444)
```

The data set name parameter (DSN) was discussed in the previous section. The SPACE and DCB parameters shown above should generally be adequate. Recall that once the data set is established, it will be assigned to a certain volume (disk pack) by the IBM operating system. Subsequent references to the data set must include this volume number in the DD statement, that is, VØL=SER=??.

Results of using a preliminary version of the graphical display package are shown in figures 28 and 29. The time-series plot shown in figure 28 was made on the line printer and the contour map shown in figure 29 was made on a CALCOMP plotter. Documentation on the use of the graphical display package is currently being written.

Modification of program logic

Some users may wish to compile only one or two numerical options with the program. This is done by removing the SOLVE routine(s) not needed from the card deck and modifying the main program in either of the following ways, assuming for this example that SIP is being removed: (1) remove the three IF statements that call SOLVE1, ITER1, and NEWITA, or (2) punch a C in column 1 of these statements and leave them in the main program.

Other modifications to the program logic will be required for certain applications. Modifications will range from changing a few statements to adding a subroutine or deleting options not used. In any case the changes should be made by a programmer familiar with the computational scheme because almost any change has an unanticipated effect on another part of the program requiring several debugging runs.

Reasonably simple modifications to the program include changing format statements and shifting data sets (for example, recharge rate) from GROUP III to GROUP IV so they can be modified for each pumping period.

Adding a second confining bed would be a more complex modification because it may require additional arrays, and ENTRY CLAY in subroutine COEF would have to be made general to accept confining-bed parameters for either bed.

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

MAIN

```
  300 CALL GRAPH   (Y(L(3)),Y(L(4)),Y(L(5)),Y(L(6)),Y(L(7)),
     1  YY(1),Y(L(8)))
      READ (R, 320,END=310) NEXT
```

STEP

```
      WRITE(10) PHI,SUM
```

PRINTAI

```
      REAL*8 HD

      DIMENSION NN(1),SUMX(1),SUMY(1),X(1),Y(1),ZZ(1),HD(1)

C***************
      ENTRY GRAPH   (SUMX,SUMY,X,Y,ZZ,HD,NN)
C***************
C  COMPUTE X AND Y COORDINATES OF ROWS AND COLUMNS
      SUMX(1)=DELX(1)/2.
      SUMY(1)=DELY(1)/2.
      DO 325 I=2,DIML
  325 SUMY(I)=SUMY(I-1)+(DELY(I)+DELY(I-1))/2.
      DO 330 I=2,DIMW
  330 SUMX(I)=SUMX(I-1)+(DELX(I)+DELX(I-1))/2.
C  DETERMINE NUMBER OF ACTIVE NODES, THEIR STORAGE LOCATION,
C  AND THEIR X AND Y COORDINATES
      N=0
      DO 340 I=2,INO1
      DO 340 J=2,JNO1
      IF(T(I,J).EQ.0.) GO TO 340
      N=N+1
      NN(N)=I+DIML*(J-1)
      X(N)=SUMX(J)
      Y(N)=SUMY(I)
  340 CONTINUE
C  WRITE X AND Y COORDINATES ON UNIT 11
      WRITE(11) (X(I),I=1,N)
      WRITE(11) (Y(I),I=1,N)
C  REWIND UNIT 10 AND REPROCESS PHI MATRIX AT EACH TIME STEP
C  PLACING PHI VALUES AT ACTIVE NODES IN THE ZZ ARRAY (REAL*4)
      REWIND 10
      DO 380 I=1,KT
      READ(10) PHI,SUM
      DO 350 J=1,N
      NIJ=NN(J)
  350 ZZ(J)=HD(NIJ)
C  WRITE PHI VALUES AT ACTIVE NODES AND ELAPSED SIMULATION TIME
C  ON UNIT 11
      WRITE(11) (ZZ(J),J=1,N),SUM
  380 CONTINUE
      WRITE(6,390) N,KT,SUMX(DIMW),SUMY(DIML)
  390 FORMAT(//,' GRAPHICS OUTPUT FOR ',I6,' ACTIVE NODES AND ',I4,
     1  ' TIME STEPS HAS BEEN WRITTEN ON UNIT 11',/,
     2  ' MAXIMUM X,Y COORDINATE PAIR IS ',F10.2,',',F10.2)
      RETURN
```

FIGURE 27.—Additional FORTRAN code required to produce output for graphical display.

```
 SAMPLE GRAPHICS OUTPUT
X,Y IS (11475.    1350.0
NML =    9
NSBH =    6
NVL =   12
NSBV =   10
NSCALE =    1
NSCALE =    0
NSCALE =    2
NSCALE =    0
NSCALE =    0

YMAX=104.00
YMIN =92.000
XMAX =.0
XMIN =.18000E 06
1 REPRESENTS  X,Y CCORDINATES (11475.    1350.0    )
2 REPRESENTS  X,Y COORDINATES (3825.0    2750.0    )
3 REPRESENTS  X,Y CCORDINATES (11475.    1950.0    )
4 REPRESENTS  X,Y COORDINATES (11475.    625.00    )
```

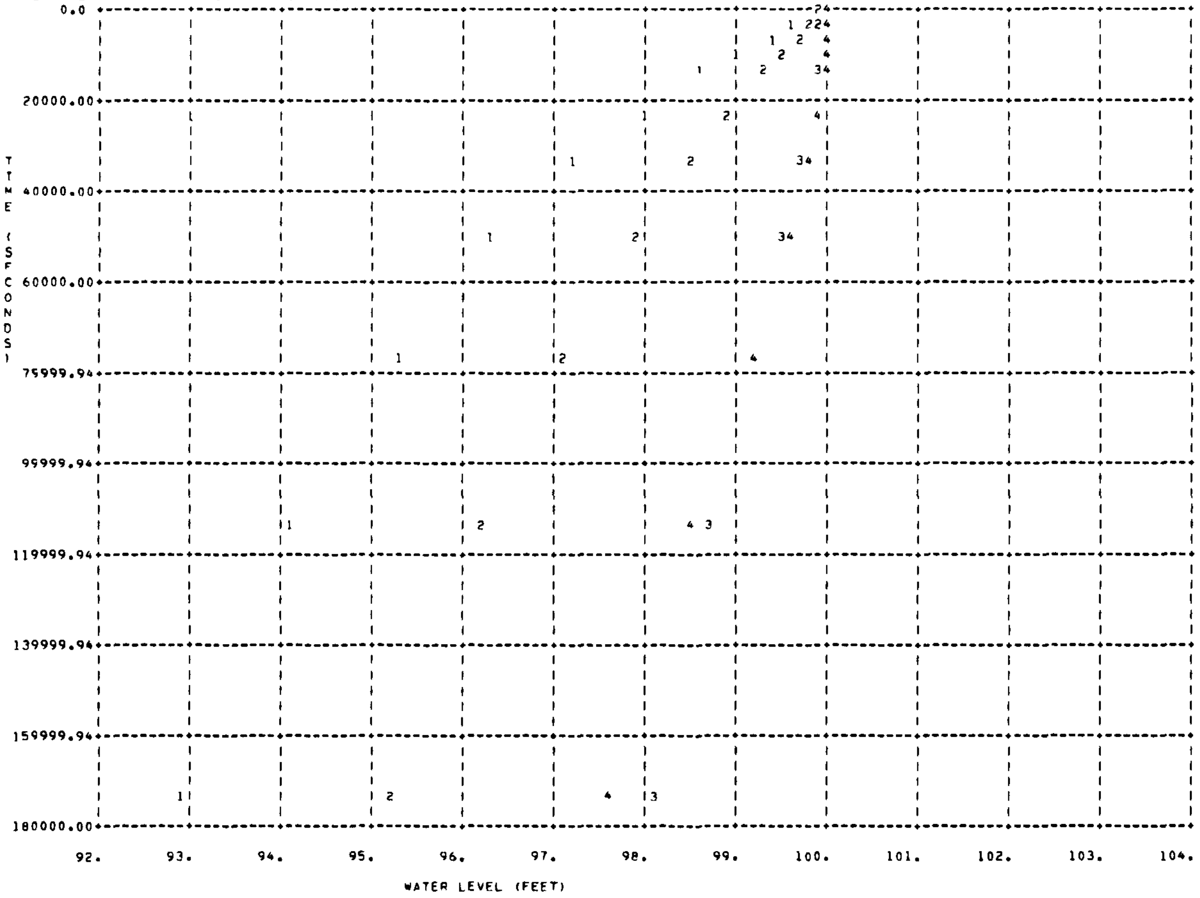


FIGURE 28.—Water level versus time at various nodes of the sample aquifer problem produced by the graphical display package.

selected installation may be available to modify the computer code as necessary.

Limitations of program

The model documented in this report is reasonably free of errors and has been used successfully to simulate a variety of aquifer systems in two dimensions. Undiscovered errors in the logic, however, may appear as the model is applied to a variety of new problems.

The user is cautioned against using this model to make more than a crude simulation of three-dimensional problems. A rigorous analysis of three-dimensional aquifer systems can be made only with the appropriate analog or digital simulators.

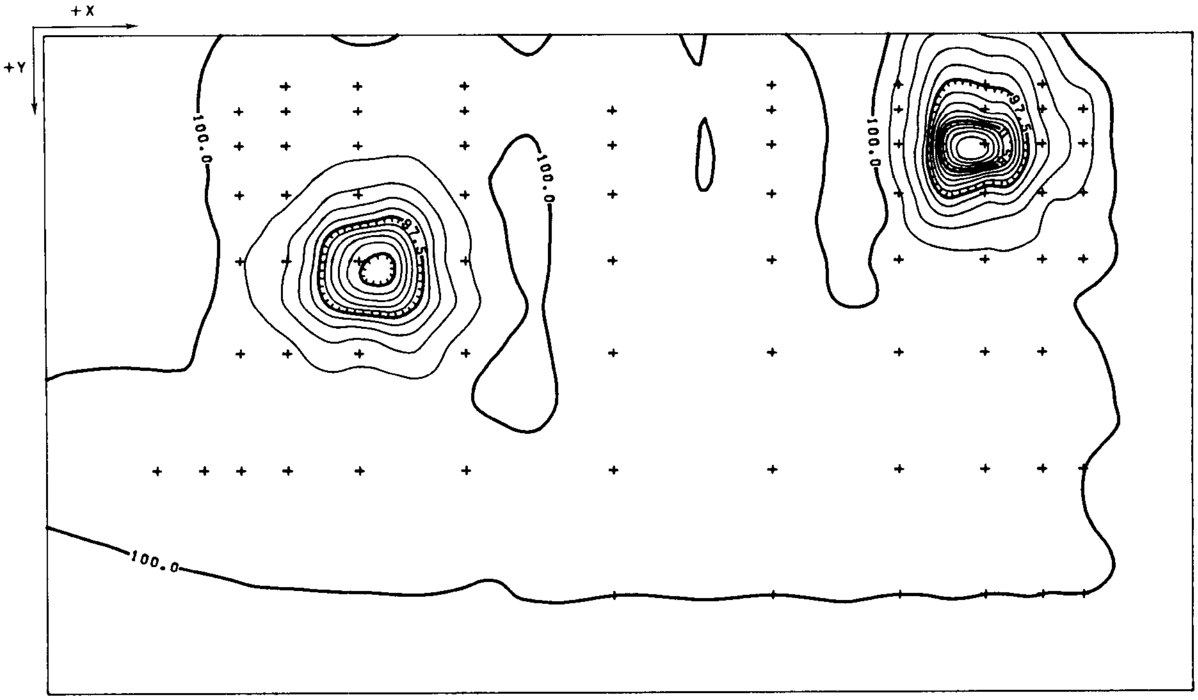


FIGURE 29.—Contour map of water level (in feet) for sample aquifer problem produced by graphical display package. Contour interval is 0.5 ft.

Attachment 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 3, punch the characters underlined in the definition, starting in the first column of the field. For any option not used, leave the appropriate columns blank.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
1	1–80	20A4	HEADNG	Any title the user wishes to print on one line at the start of output.
2	1–48	12A4		
3	1–5	A4,1X	WATER	WATE for water table or combined water-table–artesian aquifer.
	6–10	A4,1X	LEAK	LEAK for an aquifer system including leakage from a stream or confining bed.
	11–15	A4,1X	CONVRT	CONV for combined artesian–water-table aquifer.
	16–20	A4,1X	EVAP	EVAP to permit discharge by evapotranspiration.
	21–25	A4,1X	RECH	RECH to include a constant recharge rate.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
	26–30	A4,1X	NUMS	SIP or LSOR or ADI to designate the equation-solving scheme.
	31–35	A4,1X	CHCK	CHEC to compute a mass balance.
	36–40	A4,1X	PNCH	PUNC for punched output at the end of the simulation.
	41–45	A4,1X	IDK1	DK1 to read initial head and mass balance parameters from disk (unit 4).
	46–50	A4,1X	IDK2	DK2 to save (write) computed head, elapsed time, and mass balance parameters on disk (unit 4).
	51–55	A4,1X	NUM	NUME to print drawdown in numeric form.
	56–60	A4,1X	HEAD	HEAD to print the head matrix.
			(All variables on card 4 are integers)	
4	1–10	I10	DIML	Number of rows.
	11–20	I10	DIMW	Number of columns.
	21–30	I10	NW	Number of pumping wells for which drawdown is to be computed at a "real" well radius.
	31–40	I10	ITMAX	Maximum number of iterations per time step.

NOTE.—Steady-state simulations often require more than 50 iterations. Transient time steps usually require less than 30 iterations.

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 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. *Default typing of variables applies.*

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
1	1–4	A4	CONTR	CONT to generate a map of drawdown and (or) hydraulic head; *for no maps* insert a blank card.
	11–20	G10.0	XSCALE	Factor to convert model length unit to unit used in X direction on maps (that is, to convert from feet to miles, XSCALE=5280).
	21–30	G10.0	YSCALE	Factor to convert model length unit to unit used in Y direction on maps.
	31–40	G10.0	DINCH	Number of map units per inch.
	41–50	G10.0	FACT1	Factor to adjust value of drawdown printed*.
	51–60	G10.0	FACT2	Factor to adjust value of head printed*.

*Value of drawdown or head	FACT 1 or FACT 2	Printed value
52.57	.01	0
	.1	5
	1	52
	10	525
	100	***

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
	61–68	A8	MESUR	Name of map length unit.
2	1–10	G10.0	NPER	Number of pumping periods for this 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 criterion 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 reached a solution for the time step.

	31–40	G10.0	EROR	Steady-state error criterion (L).

NOTE.—If the head change between time steps in transient simulations is less than this amount, the pumping period is terminated.

	41–50	G10.0	SS	Specific storage of confining bed ($1/L$).

NOTE.—SS has a finite value only in transient simulations where leakage is a function of storage in the confining bed.

	51–60	G10.0	QET	Maximum evapotranspiration rate (L/T).
	61–70	G10.0	ETDIST	Depth at which ET ceases below land surface (L).

NOTE.—QET and ETDIST required only for simulations including evapotranspiration.

	71–80	G10.0	LENGTH	Definition depends on the numerical solution used: *LSOR:* number of LSOR iterations between 2–D corrections. *ADI and SIP:* Number of iteration parameters; unless the program is modified, code 10 for SIP.
3	1–10	G10.0	HMAX	Definition depends on numerical solution used: *LSOR:* acceleration parameter. *ADI:* maximum iteration parameter. *SIP:* value of β'.

NOTE.—See the discussion of the numerical methods in the text for information on iteration parameters.

	11–20	G10.0	FACTX	Multiplication factor for transmissivity in X direction.
	21–30	G10.0	FACTY	Multiplication factor for transmissivity in Y direction.

NOTE.—FACTX = FACTY = 1 for isotropic aquifers.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
4	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 from punched output, remove the three blank cards and insert the first three cards of the punched output from the previous run. If continuation is from interim storage on disk, the three blank cards should remain.
	21–40	G20.10	SUMP	
	41–60	G20.10	PUMPT	
	61–80	G20.10	CFLUXT	
5	1–20	G20.10	QRET	
	21–40	G20.10	CHST	
	41–60	G20.10	CHDT	
	61–80	G20.10	FLUXT	
6	1–20	G20.10	STORT	
	21–40	G20.10	ETFLXT	
	41–60	G20.10	FLXNT	

Group III: Array data

Each of the following data sets, except the first one (PHI), consists of a parameter card and, if the data set contains variable data, may include a set of data cards. *Default typing applies except for* M(I,J) *which is a real array.* Each parameter card contains five variables defined as follows:

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
Every parameter card.	1–10	G10.0	FACT	If IVAR=0, FACT is the value assigned to every element of the matrix; If IVAR=1, FACT is the multiplication factor for the following set of data cards.
	11–20	G10.0	IVAR	=0 if no data cards are to be read for this matrix; =1 if data cards for this matrix follow.
	21–30	G10.0	IPRN	=0 if input data for this matrix are to be printed; =1 if input data for the matrix are *not* to be printed.
	31–40	G10.0	IRECS	=0 if the matrix is being read from cards or if each element is being set equal to FACT. =1 if the matrix is to be read from disk (unit 2).
	41–50	G10.0	IRECD	=0 if the matrix is *not* to be stored on disk. =1 if the matrix being read from cards or set equal to FACT *is* to be stored on disk (unit 2) for later retrieval.

Refer to the examples in figures 31–33, Attachment IV. Figure 33 illustrates data for the sample problem without using disk files.

For the uniform starting head=100, FACT=100, IVAR=IPRN=IRECS=IRECD=0 and no data cards are required. The storage coefficient matrix is used to locate a constant-head boundary; therefore, FACT=−1, IVAR=1, IPRN=IRECS=IRECD=0 and a set of data cards with the location of the boundary nodes follows.

To save the storage coefficient matrix on disk (provided unit 2 has been defined on a DD statement; see technical information), set FACT = 1, IVAR = 1, IPRN = IRECS = 0, IRECD = 1, and include the set of data cards (figure 31). After this has been processed successfully, subsequent runs need only include a parameter card with the following: FACT = IVAR = IPRN = 0, IRECS = 1, IRECD = 0. The set of data cards are not included and the storage coefficient matrix is input via unit 2 from disk storage. (See figure 32.)

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 vertical conductivity of the confining bed (RATE) ranges from 2×10^{-9} to 9×10^{-8} ft/sec, coded values should range from 2 to 90; the multiplication factor (FACT) would be 1.0 E−9.

Arrays needed in every simulation are underlined. *Omit* parameter cards and data cards *not* used in the simulation (however, see the footnote for the S matrix).

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
1	1–80	8F10.4	PHI(I,J)	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.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
2	1–80	8F10.4	STRT(I,J)	Starting head matrix (*L*).
3	1–80	20F4.0	S(I,J)	Storage coefficient (dimensionless).

NOTE.—Always required. In addition to specifying storage coefficient values for artesian aquifers, this matrix is used to locate constant-head boundaries by coding a negative number at constant-head nodes. At these nodes T or PERM must be greater than zero. For a problem with no constant-head nodes and that does not require S values, insert a blank parameter card.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
4	1–80	20F4.0	T(I,J)	Transmissivity (L^2/T).

NOTE.—(1) Required for artesian aquifer simulation only.
(2) Zero values must be placed around the perimeter of the T or PERM matrix for reasons inherent in the computational scheme. If IVAR = 0, zero values are automatically inserted around the border of the model.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
5	1–80	20F4.0	PERM(I,J)	Hydraulic conductivity (*L/T*) (see note 2 for data set 4).
6	1–80	20F4.0	BOTTOM(I,J)	Elevation of bottom of aquifer (*L*).
7	1–80	20F4.0	SY(I,J)	Specific yield (dimensionless).

NOTE.—Data sets 5, 6, and 7 are required for water table or combined artesian-water table simulations.

CARD	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
8	1–80	20F4.0	TOP (I,J)	Elevation of top of aquifer (*L*).

NOTE.—Required only in combined artesian–water-table simulations.

DATA SET	*COLUMNS*	*FORMAT*	*VARIABLE*	*DEFINITION*
9	1–80	20F4.0	RATE (I,J)	Hydraulic conductivity of confining bed (*L*/*T*).
10	1–80	20F4.0	RIVER (I,J)	Head on the other side of confining bed (*L*).
11	1–80	20F4.0	M (I,J)	Thickness of confining bed (*L*).

NOTE.—Data sets 9, 10, and 11 are required in simulations with leakage. If the confining bed or streambed does not extend over the entire aquifer use the M matrix to locate the confining bed. If RATE and RIVER do not vary over the extent of the confining bed they can be initialized to a uniform value.

12	1–80	20F4.0	GRND (I,J)	Land elevation (*L*).

NOTE.—Required for simulations with evapotranspiration.

13	1–80	20F4.0	QRE (I,J)	Recharge rate (*L*/*T*).

NOTE.—Omit if not used.

14	1–80	8G10.0	DELX (J)	Grid spacing in X direction (*L*).
15	1–80	8G10.0	DELY (I)	Grid spacing in Y 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.
2. To simulate a given pumping period, set NUMT larger than the number required for the simulation period (for example, 100). The program will compute the exact DELT (which will be ≦DELT coded) and NUMT to arrive exactly at TMAX on the last time step.

Default typing applies.

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.—In general KPM1=0 if KP=1
KPM1=1 if KP=2, etc.

This causes the time parameter used in ENTRY CLAY to be set to zero and STRT to be initialized to PHI. However, for continuation of a previous pumping period KPM1=KP, and STRT and the time parameter are not affected.

	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.

	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	1	Row location of well.
11–20	G10.0	J	Column location of well.
21–30	G10.0	WELL (I,J)	Pumping rate (L^3/T), negative for a pumping well.
31–40	G10.0	RADIUS	Real well radius (L).

NOTE.—Radius is required only for those wells, if any, where computation of drawdown at a real well radius is to be made.

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

If another simulation is included in the same job, insert a blank card before the next group I cards.

Attachment IV

Sample Aquifer Simulation And Job Control Language

This appendix includes examples of job control language (JCL) for several different runs and an example problem designed to illustrate many of the options in the program. The grid and boundary conditions for the problem are given in figure 25. Figure 30 illustrates in cross section the type of problem being simulated, but note that it is not to scale.

The listing of data with the JCL examples is not on a coding form, but it should not be difficult to determine the proper location of the numbers since the fields are either 4 or 10 spaces. Zero values have not been coded on the data cards to avoid unnecessary punching.

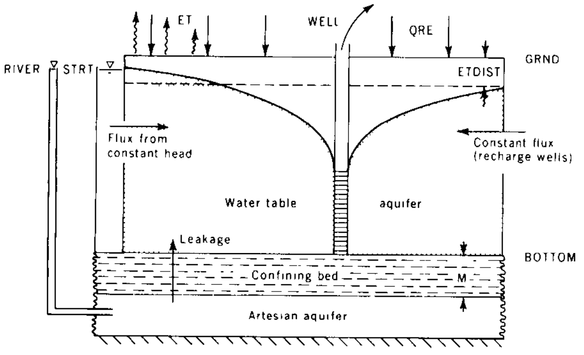


FIGURE 30.—Cross section illustrates several options included in the sample problem and identifies the meaning of several program parameters.

Figures 31 and 32 illustrate the JCL and data decks for two successive simulations of the sample problem. They are designed to show the use of disk facilities to store array data and interim results. The first run (fig. 31) is terminated after 5 iterations and interim results are stored on the data set specified by the FT04F001 DD statement. Note that arrays S, PERM, DELX, and DELY have been stored in the array data set specified by the FT02F001 DD statement (a 1 appears in column 40 of the parameter card for these arrays). The second run (fig. 32) continues computations from the previous stopping point and calculates a solution. Note that PHI, S, PERM, DELX, and DELY are read from disk storage. The final example (fig. 33) illustrates the JCL and data deck for a run without using the disk files. Following figure 33 is the output for the sample prob-

```
                 //   EXEC FORTGCG
                 //FORT.SYSIN DD *
                      Model
                      source
                      cards
                 /*
                 //GO.FT02F001 DD DSN=A442702.AZ100.AG4W0000.MATRIX,
                 //     UNIT=ONLINE,DISP=(NEW,KEEP),
                 //     SPACE=(560,(14)),DCB=(RECFM=F)
                 //GO.FT04F001 DD DSN=A442702.AZ100.AG4W0000.HEAD,
                 //     UNIT=ONLINE,DISP=(NEW,KEEP),SPACE=(TRK,(1,1),RLSE),
                 //     DCB=(RECFM=VBS,LRECL=1168,BLKSIZE=1172)
                 //GO.SYSIN DD *
                                                                  ---- SAMPLE AQUIFER PROBLEM ---
                 -
Group I          WATE LEAK      EVAP RECH SIP  CHEC           DK2  NUME HEAD
                         10        14         1         5
                 CONT               1         1      1500         1        .1  FEET
Group II                  1         1      .003       .01         0    .4E-06        10        10
                          1         1         1

          STRT          100
                         -1         1                             1
          S

                                                                    1
                                                5   1   1   1   1   1
                       .002         1                             1
                                    1   1   1       2   2   2   2
                                1   1   1   1   2   2   2   2   2   2
Group III PERM                  1   1   1   2   2   2   2   2   2   2
                                2   2   2   2   2   2   2   2   2   2
                                2   2   2   3   3   3   3   3   3   3
                                3   3   3   3   4   4   4   4   4
                        4   4   4   4   4   4   4   3   3   3   3   3
                                                3   3   3   3   3   3
         BOTTOM           0
          SY
          RATE       .3E-07
          RIVER         100
          M              10
          GRND          105
          QRE        .2E-07
                         50         1                             1
          DELX           20        14         9         9        14        21        31        41
                         37        25        17        11         9        13
                         50         1                             1
          DELY           10         5         7        10        14        18        27        30
                         31        12
                          1         0         6         1         1       1.0        24
                          4         4       .05
                          5         4       .05
Group IV                  6         4       .05
                          7         4       .05
                          4        11       -10         2
                          6         6       -10
                 /*
```

FIGURE 31.—JCL and data deck to copy some of the data sets on disk, compute for 5 iterations, and store the results on disk.

```
//   EXEC FORTGCG
//FORT.SYSIN DD *
```

Model source cards

```
/*
//GO.FT02F001 DD DSN=A442702.AZ100.AG4W0000.MATRIX,
//    UNIT=ONLINE,DISP=SHR,VOL=SER=SYS015
//GO.FT04F001 DD DSN=A442702.AZ100.AG4W0000.HEAD,
//    UNIT=ONLINE,DISP=SHR,VOL=SER=SYS011
//GO.SYSIN DD *
```

Group I

```
                                                ---- SAMPLE AQUIFER PROBLEM ---
-
WATE LEAK      EVAP REOH SIP  CHEC      DK1       NUME HEAD
        10        14         1        50
```

Group II

```
CONT               1         1      1500         1        .1  FEET
         1         1      .003       .01         0    .4E-06        10        10
         1         1         1
```

Group III

```
STRT           100
S                                    1         1
PERM                                 1         1
BOTTOM           0
SY
RATE        .3E-07
RIVER          100
M               10
GRND           105
QRE         .2E-07
DELX                                 1         1
DELY                                           1
```

Group IV

```
         1         0         6         1         1       1.0        24
         4         4       .05
         5         4       .05
         6         4       .05
         7         4       .05
         4        11       -10         2
         6         6       -10
```

```
/*
//
```

FIGURE 32.—JCL and data deck to continue the previous run (fig. 31) to a solution.

lem generated using the JCL and problem deck shown in figure 33.

Figures 31 to 33 show that the source cards are being compiled for each run. It is more efficient, of course, to compile the source deck once and store it as a load module on disk. Subsequent runs can use the load module with considerable reduction in cards read, CPU time, and lines printed.

```
                  //   EXEC FORTGCG
                  //FORT.SYSIN DD *
                  +----------------+
                  |     Model      |
                  |     source     |
                  |     cards      |
                  +----------------+
                  /*
                  //GO.SYSIN DD *
                                                                   ---- SAMPLE AQUIFER PROBLEM ---
Group I           -
                  WATE LEAK      EVAP RECH SIP  CHEC                NUME HEAD
                          10        14         1        50
                  CONT               1         1      1500         1        .1  FEET
Group II                   1         1      .003       .01         0    .4E-06        10        10
                           1         1         1

          STRT           100
                          -1         1







          S                                                        1
                                              1   1   1   1   1   1
                        .002         1

                                        1   1   1       2   2   2   2
                                    1   1   1   1   2   2   2   2   2   2
Group III PERM                      1   1   1   2   2   2   2   2   2   2
                                    2   2   2   2   2   2   2   2   2   2
                                    2   2   2   3   3   3   3   3   3   3
                                    3   3   3   3   4   4   4   4   4
                            4   4   4   4   4   4   4   3   3   3   3   3
                                                    3   3   3   3   3   3
          BOTTOM           0
          SY
          RATE       .3E-07
          RIVER          100
          M               10
          GRND           105
          QRE        .2E-07
                          50         1
          DELX            20        14         9         9        14        21        31        41
                          37        25        17        11         9        13
                          50         1
          DELY            10         5         7        10        14        18        27        30
                          31        12
                           1         0         6         1         1       1.0        24
                           4         4       .05
Group IV                   5         4       .05
                           6         4       .05
                           7         4       .05
                           4        11       -10         2
                           6         6       -10
                  /*
```

FIGURE 33.—JCL and data deck to simulate the sample problem without using disk files.

Program Output using data deck illustrated in figure 33

```
                                                            U. S. G. S.

                                                      FINITE-DIFFERENCE MODEL
                                                                FOR
                                                  SIMULATION OF GROUND-WATER FLOW

                                                           JANUARY, 1975
************************************************************************************************************************************

                                                 ---- SAMPLE AQUIFER PROBLEM ----

************************************************************************************************************************************

SIMULATION OPTIONS:  WATE    LEAK            EVAP    RECH    SIP     CHEC                            NUME    HEAD

                                                              NUMBER OF ROWS =   10
                                                           NUMBER OF COLUMNS =   14
        NUMBER OF WELLS FOR WHICH DRAWDOWN IS COMPUTED AT A SPECIFIED RADIUS =    1
                                      MAXIMUM PERMITTED NUMBER OF ITERATIONS =   50

                                                      WORDS OF Y VECTOR USED =   3731

                              ON ALPHAMERIC MAP:
                                       MULTIPLICATION FACTOR FOR X DIMENSION =   1.000000
                                       MULTIPLICATION FACTOR FOR Y DIMENSION =   1.000000
                                                      MAP SCALE IN UNITS OF       FEET
                                                 NUMBER OF   FEET   PER INCH =   1500.000
                                          MULTIPLICATION FACTOR FOR DRAWDOWN =   1.000000
                                              MULTIPLICATION FACTOR FOR HEAD =  0.9999996E-01

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

                                                  ERROR CRITERIA FOR CLOSURE =  0.3000000E-02
                                                 STEADY STATE ERROR CRITERIA =  0.9999998E-02

                                           SPECIFIC STORAGE OF CONFINING BED =  0.0
                                                     EVAPOTRANSPIRATION RATE =  0.4000000E-06
                                                       EFFECTIVE DEPTH OF ET =   10.00000

                     MULTIPLICATION FACTOR FOR TRANSMISSIVITY IN X DIRECTION =   1.000000
                                                              IN Y DIRECTION =   1.000000

                                                               STARTING HEAD =   100.0000
```

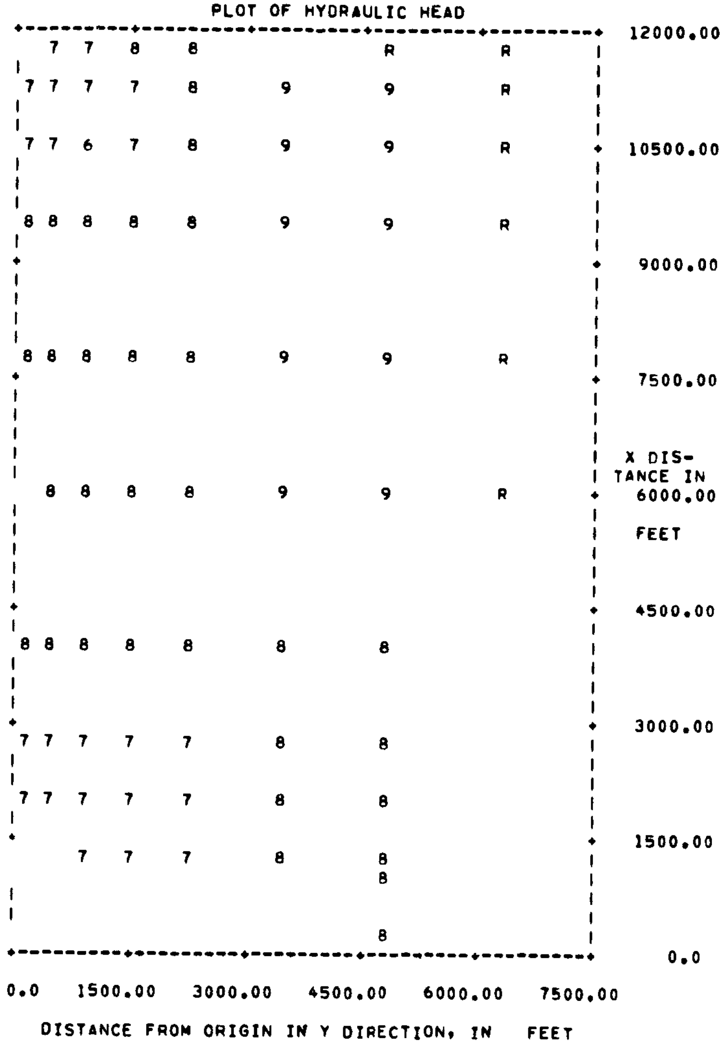
```
                                                   STORAGE COEFFICIENT
                                                         MATRIX
                                          -----------------------------------
  1  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  2  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  3  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  4  0.0       0.0       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       0.0       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       0.0       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       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  8  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0     -1.00000    0.0
  9  0.0       0.0       0.0       0.0       0.0       0.0       0.0     -1.00000  -1.00000  -1.00000  -1.00000  -1.00000  -1.00000    0.0
 10  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
```

```
                                           AQUIFER HYDRAULIC CONDUCTIVITY
                                                       MATRIX
                                         ------------------------------------

  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

  2 0.0         0.0         0.0         0.0         0.200E-02   0.200E-02   0.200E-02   0.0         0.400E-02   0.400E-02
    0.400E-02   0.400E-02   0.0         0.0

  3 0.0         0.0         0.0         0.200E-02   0.200E-02   0.200E-02   0.200E-02   0.400E-02   0.400E-02   0.400E-02
    0.400E-02   0.400E-02   0.400E-02   0.0

  4 0.0         0.0         0.0         0.200E-02   0.200E-02   0.200E-02   0.400E-02   0.400E-02   0.400E-02   0.400E-02
    0.400E-02   0.400E-02   0.400E-02   0.0

  5 0.0         0.0         0.0         0.400E-02   0.400E-02   0.400E-02   0.400E-02   0.400E-02   0.400E-02   0.400E-02
    0.400E-02   0.400E-02   0.400E-02   0.0

  6 0.0         0.0         0.0         0.400E-02   0.400E-02   0.400E-02   0.600E-02   0.600E-02   0.600E-02   0.600E-02
    0.600E-02   0.600E-02   0.600E-02   0.0

  7 0.0         0.0         0.0         0.600E-02   0.600E-02   0.600E-02   0.600E-02   0.800E-02   0.800E-02   0.800E-02
    0.800E-02   0.800E-02   0.0         0.0

  8 0.0         0.800E-02   0.800E-02   0.800E-02   0.800E-02   0.800E-02   0.800E-02   0.800E-02   0.600E-02   0.600E-02
    0.600E-02   0.600E-02   0.600E-02   0.0

  9 0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.600E-02   0.600E-02   0.600E-02
    0.600E-02   0.600E-02   0.600E-02   0.0

 10 0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
    0.0         0.0         0.0         0.0
                                         AQUIFER BASE ELEVATION        =  0.0

                                            SPECIFIC YIELD             =  0.0

                                     CONFINING BED HYDRAULIC CONDUCTIVITY=  0.3000000E-07

                                               RIVER HEAD              =   100.0000

                                         CONFINING BED THICKNESS       =   10.00000

                                         LAND SURFACE ELEVATION        =   105.0000

                                           AREAL RECHARGE RATE         =  0.2000000E-07
```

```
                                        GRID SPACING IN PROTOTYPE IN X DIRECTION
                                        ----------------------------------------

  1000.      700.      450.      450.      700.     1050.     1550.     2050.     1850.     1250.      850.      550.
   450.      650.

                                        GRID SPACING IN PROTOTYPE IN Y DIRECTION
                                        ----------------------------------------

   500.      250.      350.      500.      700.      900.     1350.     1500.     1550.      600.

                                      SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE
                                      ___________________________________________

                                              BETA= 1.00

10 ITERATION PARAMETERS: 0.0            0.6903903D 00  0.9041418D 00  0.9703214D 00  0.9908112D 00. 0.0
                         0.6903903D 00  0.9041418D 00  0.9703214D 00  0.9908112D 00

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

                                              NUMBER OF TIME STEPS=     1

                                                    DELT IN HOURS =    24.000

                                              MULTIPLIER FOR DELT =     1.000

                                                          6 WELLS
                                                        ---------

                                          I         J    PUMPING RATE  WELL RADIUS

                                          4         4        0.05
                                          5         4        0.05
                                          6         4        0.05
                                          7         4        0.05
                                          4        11      -10.00          2.00
                                          6         6      -10.00
```

```
                                        ----------------------------------------------------------
                                        I               TIME STEP NUMBER =        1              I
                                        ----------------------------------------------------------

                                             SIZE OF TIME STEP IN SECONDS=      86400.00

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

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


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

                         SOURCES:                                                              STORAGE =             0.0
                         --------                                                             RECHARGE =             1.1873
                                STORAGE =                 0.0                            CONSTANT FLUX =             0.2000
                               RECHARGE =           102586.63                                  PUMPING =           -20.0000
                          CONSTANT FLUX =            17279.99                       EVAPOTRANSPIRATION =            -0.2412
                          CONSTANT HEAD =          1418633.00                           CONSTANT HEAD:
                                LEAKAGE =           210060.38                                       IN =            16.4194
                          TOTAL SOURCES =          1748559.00                                      OUT =             0.0
                                                                                          LEAKAGE:
                         DISCHARGES:                                         FROM PREVIOUS PUMPING PERIOD =          0.0
                         -----------                                                             TOTAL =             2.4313
                     EVAPOTRANSPIRATION =            20837.56
                          CONSTANT HEAD =                 0.0                           SUM OF RATES =              -0.0032
                        QUANTITY PUMPED =          1727998.00
                                LEAKAGE =                 0.0
                        TOTAL DISCHARGE =          1748835.00

                      DISCHARGE-SOURCES =              276.00
                     PER CENT DIFFERENCE =               0.02


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

              9.5204      4.8325      3.7815      7.4434      3.4337      2.6980      1.3149      1.8210      1.1354      0.8168

              0.4495      0.5055      0.3512      0.3693      0.2810      0.2107      0.0960      0.1267      0.0765      0.0509

              0.0297      0.0322      0.0225      0.0234      0.0179      0.0133      0.0060      0.0080      0.0048      0.0032

              0.0019

          MAXIMUM CHANGE IN HEAD FOR THIS TIME STEP =     32.067
          ----------------------------------------------------

          TIME STEP :  1
          ----------

          ITERATIONS: 30
```

```
                          PLOT OF DRAWDOWN
+----------+----------+----------+----------+----------+----------+  12000.00
  23 23  19    16                          R          R          I
I
2525 24  20    15        9                 1          R          I
I                                                                I
I
2728 32  21    14        9                 3          R          +  10500.00
I                                                                I
I                                                                I
I
1919 18  16    12        8                 4          R          I
I                                                                I
+                                                                +   9000.00
I                                                                I
I                                                                I
I                                                                I
1413 13  12    10        8                 4          R          I
+                                                                +   7500.00
I                                                                I
I                                                                I
I                                                                I  X DIS-
I                                                                I TANCE IN
  13 13  12    11        9                 6          R          +   6000.00
I                                                                I
I                                                                I   FEET
I                                                                I
I                                                                I
+                                                                +   4500.00
I                                                                I
1717 17  17    16       14                12                     I
I                                                                I
I                                                                I
+                                                                +   3000.00
2020 21  22    27       18                15                     I
I                                                                I
I
2021 21  22    22       18                15                     I
I                                                                I
+                                                                +   1500.00
  21 21  21    21       18                16                     I
                                          16                     I
I                                                                I
I                                                                I
I                                         16                     I
+----------+----------+----------+----------+----------+----------+  0.0

0.0    1500.00    3000.00    4500.00    6000.00    7500.00

   DISTANCE FROM ORIGIN IN Y DIRECTION, IN    FEET

EXPLANATION
-----------

R = CONSTANT HEAD BOUNDARY
*** = VALUE EXCEEDED 3 FIGURES
MULTIPLICATION FACTOR =    1.000
```



```
                      PLOT OF HYDRAULIC HEAD
+---------+---------+---------+---------+---------+  12000.00
   7  7   8    8                R         R       I
I
I7 7  7   7    8       9        9         R       I
I                                                 I
I7 7  6   7    8       9        9         R       +  10500.00
I                                                 I
I                                                 I
I                                                 I
I8 8  8   8    8       9        9         R       I
I                                                 I
+                                                 +   9000.00
I                                                 I
I                                                 I
I                                                 I
I                                                 I
I8 8  8   8    8       9        9         R       I
+                                                 +   7500.00
I                                                 I
I                                                 I
I                                                 I
I                                                 I  X DIS-
                                                  I TANCE IN
   8  8   8    8       9        9         R       +  6000.00
I                                                 I
I                                                 I   FEET
I                                                 I
I                                                 I
+                                                 +   4500.00
I                                                 I
I8 8  8   8    8       8        8                 I
I                                                 I
I                                                 I
I                                                 I
+                                                 +   3000.00
I7 7  7   7    7       8        8                 I
I                                                 I
I                                                 I
I7 7  7   7    7       8        8                 I
I                                                 I
+                                                 +   1500.00
      7   7    7       8        8                 I
                                8                 I
I                                                 I
I                                                 I
                                8                 I
+---------+---------+---------+---------+---------+    0.0

0.0     1500.00   3000.00   4500.00   6000.00   7500.00
   DISTANCE FROM ORIGIN IN Y DIRECTION, IN   FEET
```

```
EXPLANATION
-----------

R = CONSTANT HEAD BOUNDARY
*** = VALUE EXCEEDED 3 FIGURES
MULTIPLICATION FACTOR =    0.100
```

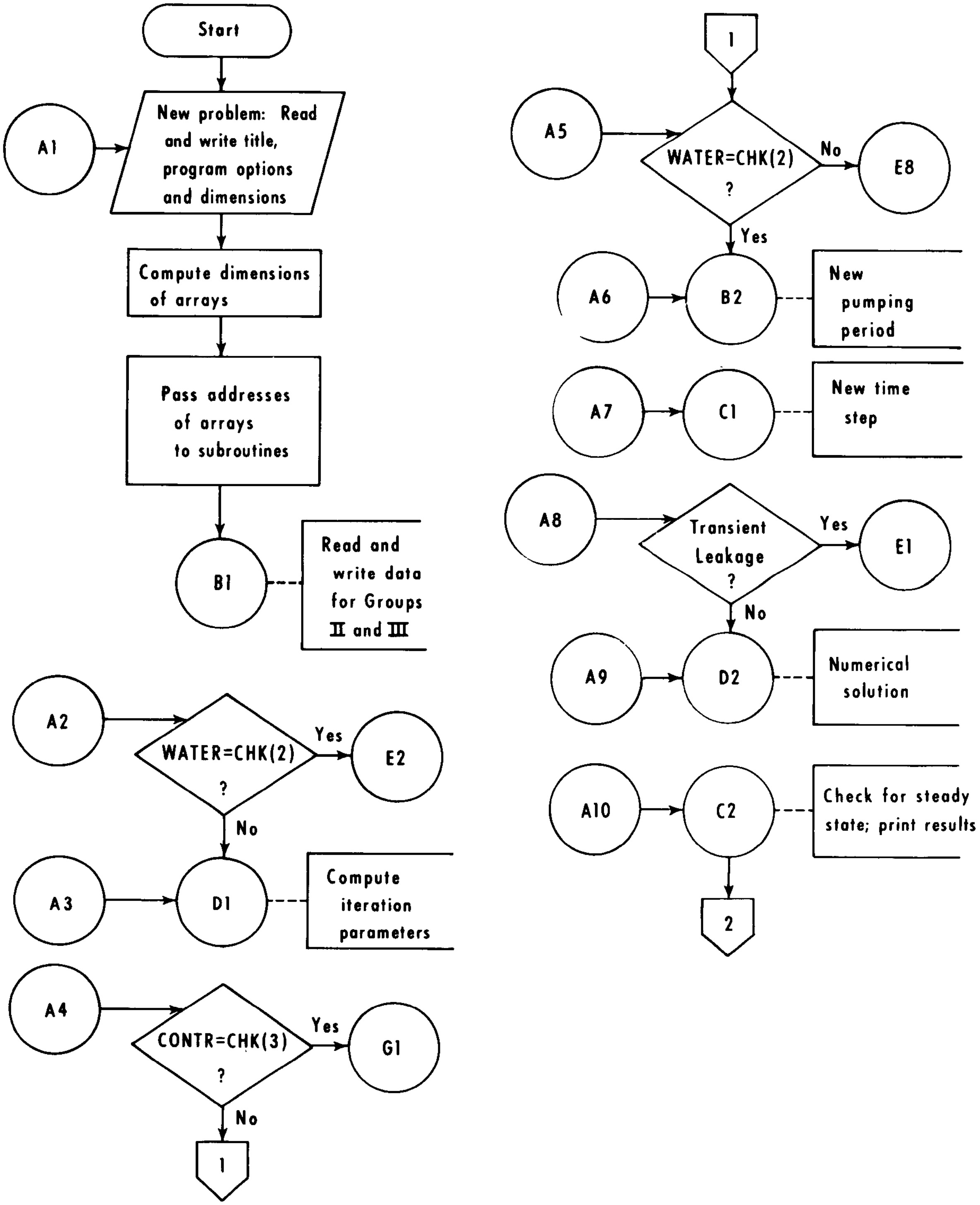
```
                                 HEAD MATRIX
                                 -----------

 1   100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
 2   100.0 100.0 100.0 100.0  79.0  79.5  82.7 100.0  85.9  80.2  72.6  74.3 100.0 100.0
 3   100.0 100.0 100.0  78.9  78.9  79.4  82.8  86.8  86.1  80.4  71.7  74.6  76.1 100.0
 4   100.0 100.0 100.0  78.8  78.6  78.7  82.8  86.9  86.5  81.1  67.9  75.3  76.9 100.0
 5   100.0 100.0 100.0  78.4  78.0  77.0  82.7  87.3  87.5  83.8  78.6  79.8  80.1 100.0
 6   100.0 100.0 100.0  78.6  77.5  72.1  83.0  88.3  89.2  87.4  85.5  84.9  84.0 100.0
 7   100.0 100.0 100.0  81.5  81.4  81.6  85.4  90.1  91.4  91.1  90.9  90.9 100.0 100.0
 8   100.0  84.0  83.9  83.9  84.0  84.8  87.6  93.4  95.1  95.7  96.6  98.1 100.0 100.0
 9   100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
10   100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0 100.0
```

```
                                        DRAWDOWN
                                        --------

 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
 2    0.0   0.0   0.0   0.0  21.0  20.5  17.3   0.0  14.1  19.8  27.4  25.7   0.0   0.0
 3    0.0   0.0   0.0  21.1  21.1  20.6  17.2  13.2  13.9  19.6  28.3  25.4  23.9   0.0
 4    0.0   0.0   0.0  21.2  21.4  21.3  17.2  13.1  13.5  18.9  32.1  24.7  23.1   0.0
 5    0.0   0.0   0.0  21.6  22.0  23.0  17.3  12.7  12.5  16.2  21.4  20.2  19.9   0.0
 6    0.0   0.0   0.0  21.4  22.5  27.9  17.0  11.7  10.8  12.6  14.5  15.1  16.0   0.0
 7    0.0   0.0   0.0  18.5  18.6  18.4  14.6   9.9   8.6   8.9   9.1   9.1   0.0   0.0
 8    0.0  16.0  16.1  16.1  16.0  15.2  12.4   6.6   4.9   4.3   3.4   1.9   0.0   0.0
 9    0.0   0.0   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   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

                        HEAD AND DRAWDOWN IN PUMPING WELLS
                        ----------------------------------

                       I    J   WELL RADIUS     HEAD     DRAWDOWN

                       4   11       2.00        35.10      64.90
```

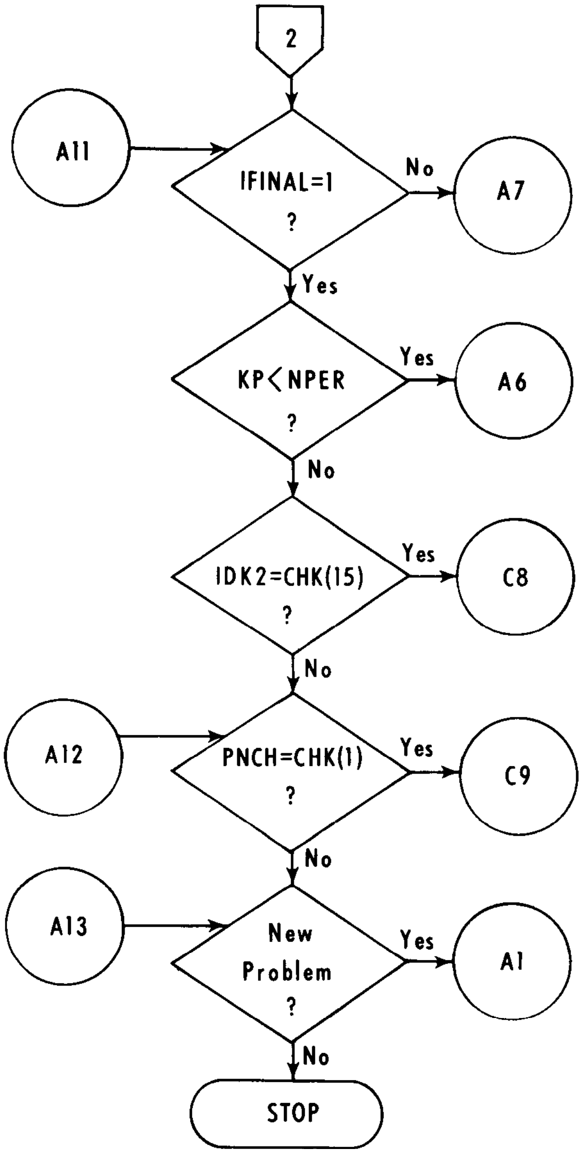
Attachment V

Generalized Flow Chart For Aquifer Simulation Model

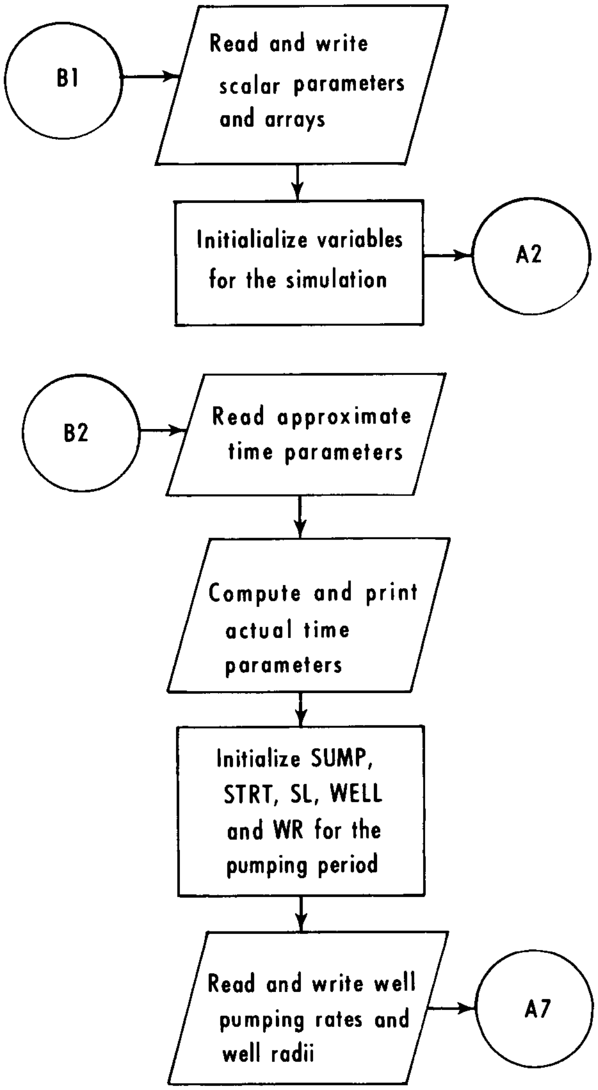
A, MAIN PROGRAM



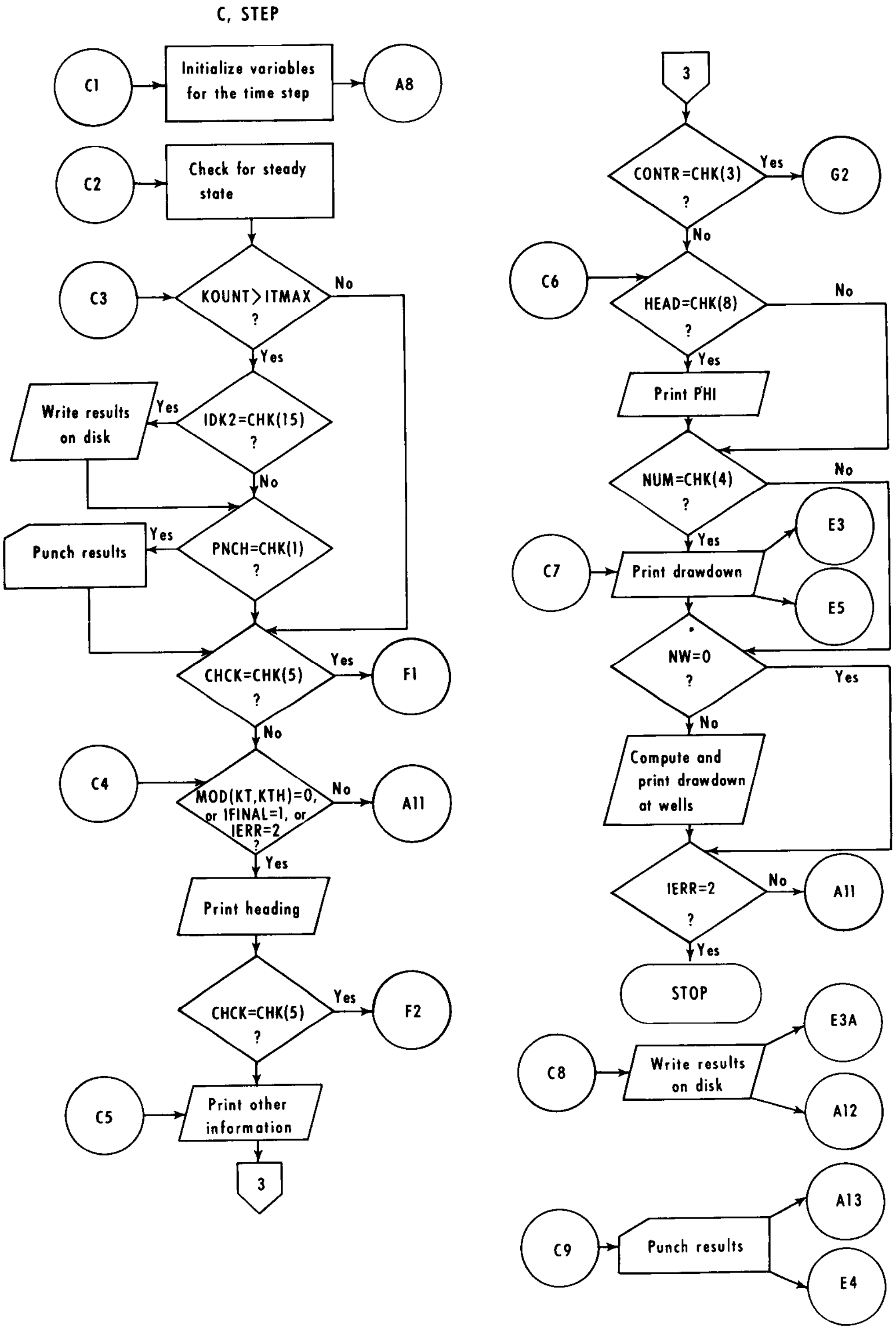
Flow chart—Continued



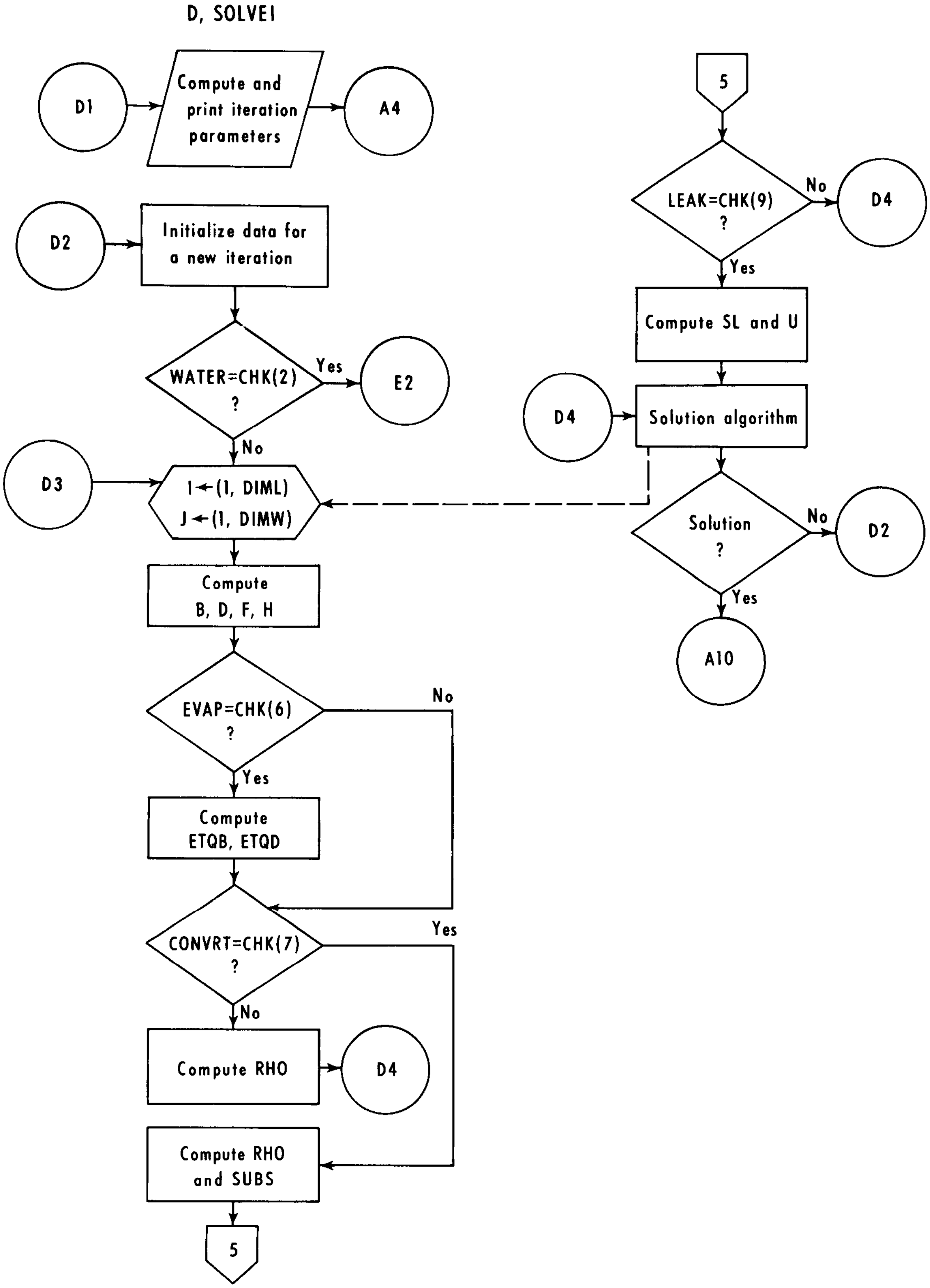
B, DATA1



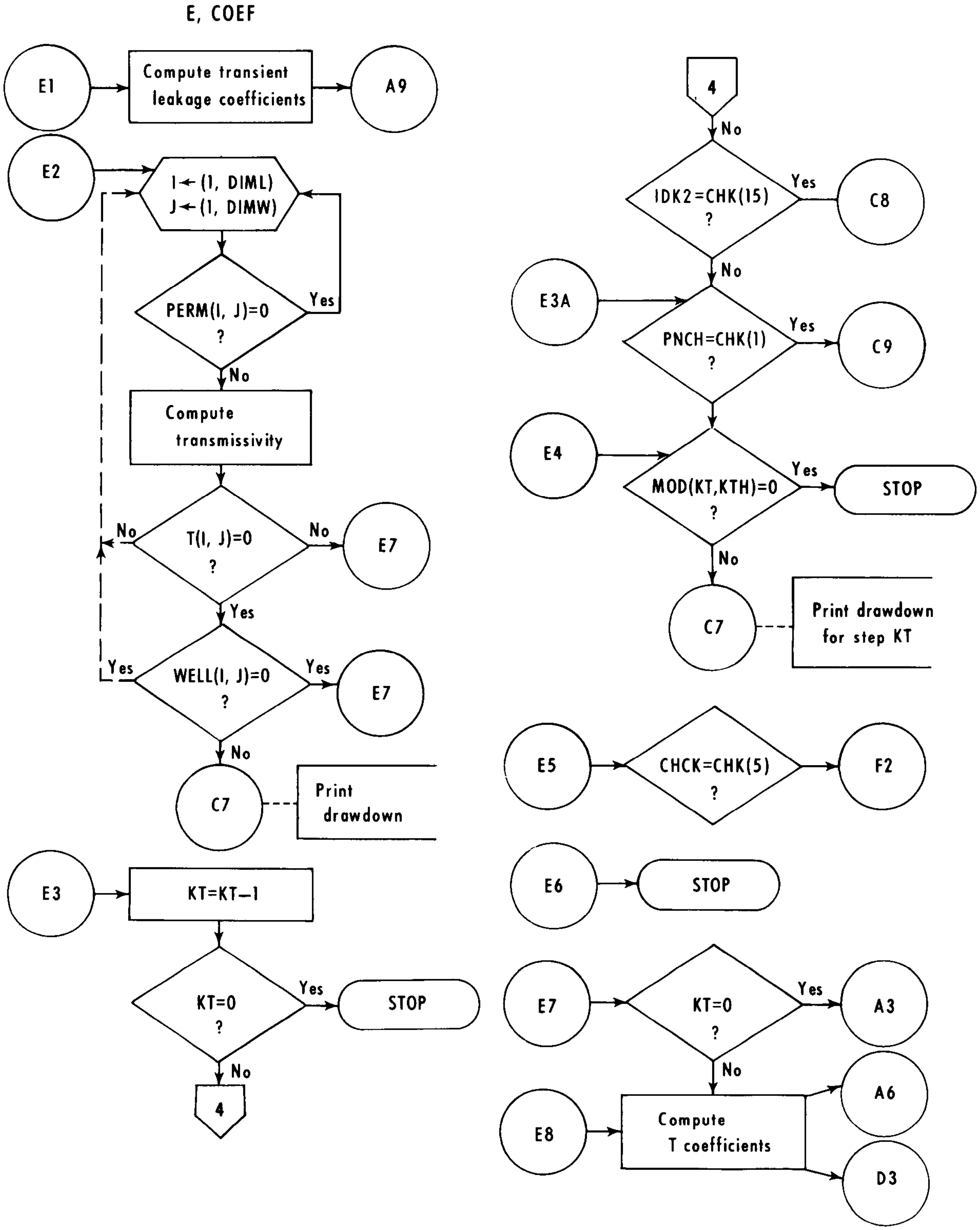
Flow chart—Continued



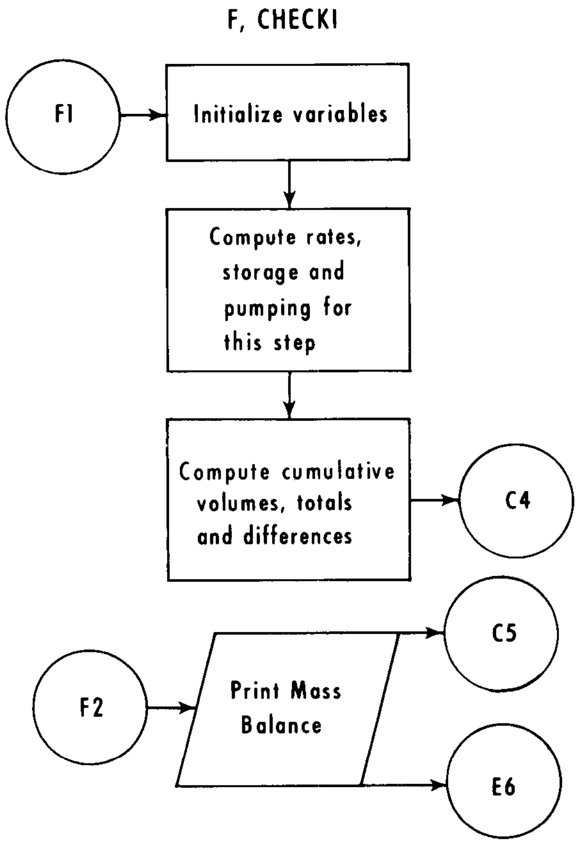
Flow chart—Continued

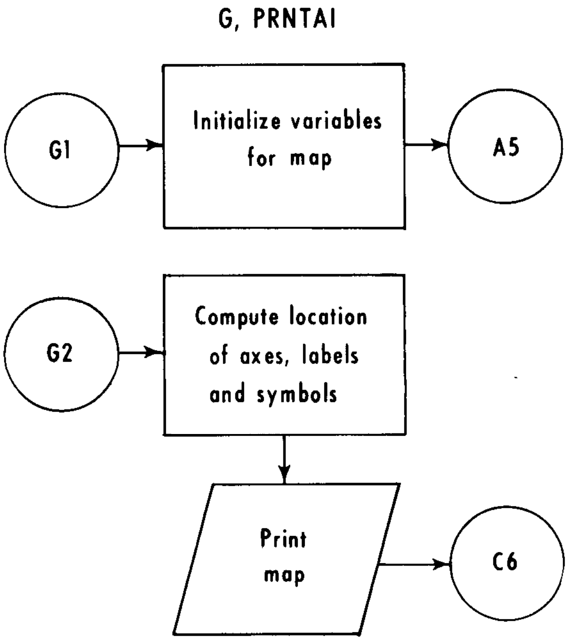


Flow chart—Continued



Flow chart—Continued





Attachment VI

Definition Of Program Variables

A	IN DATAI, DUMMY ARRAY (DOES NOT USE CORE SPACE) USED TO OBTAIN ADDRESSES OF ARRAY DATA SETS;
ALFA	CORRECTION VECTOR FOR ROWS (LSOR); PARAMETER IN SIP ALGORITHM;
B	TC(I-1,J)/DELY(I) (1/T);
BE	PARAMETER IN THOMAS ALGORITHM;
BOTTOM	ELEVATION OF THE BOTTOM OF THE AQUIFER (L);
CDLT	MULTIPLYING FACTOR FOR THE TIME STEP;
CHCK	CONTAINS CHARACTER STRING FOR MASS BALANCE OPTION;
CHK	VECTOR CONTAINING PROBLEM OPTIONS;
CONTR	CONTAINS CHARACTER STRING FOR OPTION TO PRINT MARS OF DRAWDOWN AND/OR HEAD;
CONVRT	CONTAINS CHARACTER STRING FOR WATER TABLE-ARTESIAN OPTION;
D	TR(I,J-1)/DELX(J) (1/T);
DDN	VECTOR THAT CONTAINS DRAWDOWN VALUES (L);
DEL	ARRAY USED IN SIP ALGORITHM;
DELT	TIME INCREMENT (T);
DELX	GRID SPACING IN THE X DIRECTION (L);
DELY	GRID SPACING IN THE Y DIRECTION (L);
DIML	NUMBER OF ROWS;
DIMW	NUMBER OF COLUMNS;
EROR	STEADY STATE EROR CRITERION (L);
ERR	CLOSURE CRITERION (L);
ETA	ARRAY USED IN SIP ALGORITHM;
ETDIST	DEPTH AT WHICH EVAPOTRANSPIRATION CEASES BELOW LAND SURFACE (L);
ETQB	THAT PART OF ET SOURCE TERM TREATED IMPLICITLY;
ETQD	THAT PART OF ET SOURCE TERM TREATED EXPLICITLY;
EVAP	CONTAINS CHARACTER STRING FOR EVAPOTRANSPIRATION OPTION;
F	TR(I,J)/DELX(J) (1/T);
FACT	SEE EXPLANATION IN GROUP III; ARRAY DATA;
FACTX	MULTIPLICATION FACTOR FOR TRANSMISSIVITY IN X DIRECTION;
FACTY	MULTIPLICATION FACTOR FOR TRANSMISSIVITY IN Y DIRECTION;
G	PARAMETER IN THOMAS ALGORITHM;
H	TC(I,J)/DELY(I) (1/T);
GRND	ELEVATION OF LAND SURFACE (L);
HEAD	CONTAINS CHARACTER STRING FOR OPTION TO PRINT HEAD VALUES;
HEADNG	TITLE FOR SIMULATION;
HMAX	MAXIMUM ITERATION PARAMETER (ADI); ACCELERATION PARAMETER (LSOR); BETA PARAMETER (SIP);
IC	INDICATOR USED TO DETERMINE THE TYPE OF ARRAY DATA;
IERR	= 0 PUMPING WELLS ARE IN SATURATED PART OF WATER TABLE AQUIFER; = 1 PUMPING WELL HAS GONE DRY;
IFINAL	= 0 ALL TIME STEPS EXCEPT THE LAST; = 1 LAST TIME STEP IN PUMPING PERIOD;
IFMT1,IFMT2,IFMT3	VARIABLE FORMAT ARRAYS PASSED TO DATAI VIA ARRAY ENTRY POINT;
IN	IN DATAI, DUMMY ARRAY TO WHICH NAME IS PASSED;
INO1	DIML-1;
IPRN	SEE EXPLANATION IN GROUP III; ARRAY DATA;
IRECS,IRECD	SEE EXPLANATION IN GROUP III,ARRAY DATA;
IRN	RECORD NUMBER USED FOR DISK STORAGE AND RETRIEVAL OF ARRAY DATA;

Definition of program variables—Continued

ITMAX	MAXIMUM NUMBER OF ITERATIONS PER TIME STEP;
IVAR	SEE EXPLANATION IN GROUP III; ARRAY DATA;
ISUM	THE CUMULATIVE WORDS OF STORAGE USED IN THE Y VECTOR;
IZ,JZ,ETC.	DIMENSIONS OF ARRAYS IN MODEL,COMPUTED IN MAIN PROGRAM;
JN01	DIMW-1;
KEEP	HYDRAULIC HEAD AT THE PREVIOUS TIME STEP (L);
KKK	ASSOCIATED VARIABLE IN DEFINE FILE, INDICATES NUMBER OF NEXT RECORD;
KOUNT	ITERATION COUNTER;
KP	NUMBER OF THE PUMPING PERIOD;
KPM1	NUMBER OF PREVIOUS PUMPING PERIOD;
KT	TIME STEP COUNTER;
KTH	NUMBER OF TIME STEPS BETWEEN PRINTOUTS;
L	VECTOR CONTAINING INITIAL ADDRESS OF ARRAYS;
LEAK	CONTAINS CHARACTER STRING FOR LEAKAGE OPTION;
LENGTH	NUMBER OF ITERATION PARAMETERS (SIP,ADI); NUMBER OF ITERATIONS BETWEEN 2-D CORRECTION (LSOR);
M	THICKNESS OF CONFINING OR STREAM BED (L);
NPER	NUMBER OF PUMPING PERIODS;
NUM	CONTAINS CHARACTER STRING FOR OPTION TO PRINT DRAWDOWN;
NUMT	NUMBER OF TIME STEPS;
NW	NUMBER OF PUMPING WELLS FOR WHICH DRAWDOWN IS TO BE COMPUTED AT A 'REAL' WELL RADIUS;
NWEL	NUMBER OF WELLS FOR A PUMPING PERIOD;
NWR	LOCATION OF WELLS;
PNCH	CONTAINS CHARACTER STRING FOR OPTION TO PUNCH HYDRAULIC HEAD VALUES;
P	PRINTER UNIT NUMBER;
PARAM	ITERATION PARAMETER;
PERM	HYDRAULIC CONDUCTIVITY OF THE AQUIFER (L/T);
PHE	HYDRAULIC HEAD AT THE START OF THE ITERATION (L);
PHI	HYDRAULIC HEAD (L);
PU	PUNCH UNIT NUMBER;
QET	MAXIMUM EVAPOTRANSPIRATION RATE (L/T);
QRE	RECHARGE RATE (L/T);
R	READER UNIT NUMBER;
RADIUS	REAL WELL RADIUS (L);
RATE	VERTICAL HYDRAULIC CONDUCTIVITY OF THE CONFINING BED OR STREAM BED (L/T);
RECH	CONTAINS CHARACTER STRING FOR RECHARGE OPTION;
RHO	S/DELT (1/T);
RHOP	VECTOR CONTAINING ITERATION PARAMETERS;
RIVER	HYDRAULIC HEAD OF THE STREAM OR IN THE AQUIFER ABOVE OR BELOW THE PUMPED AQUIFER (L);
RW	WELL AND RECHARGE SOURCE TERM (L/T);
S	STORAGE COEFFICIENT;
SIP	CONTAINS CHARACTER STRING FOR SIP OPTION;
SL	STEADY PART OF LEAKAGE COEFFICIENT (L/T);
SLEAK	INITIAL & TRANSIENT LEAKAGE (L/T);
SS	SPECIFIC STORAGE OF CONFINING BED (1/L);
STORE	CONTAINS EITHER THE STORAGE COEFFICIENT OR SPECIFIC YIELD DEPENDING ON THE TYPE OF AQUIFER;
STRT	HYDRAULIC HEAD AT THE BEGINNING OF THE CURRENT PUMPING PERIOD (L);
SUBS	MODIFIES STORAGE TERM IN WATER TABLE-ARTESIAN CONVERSION;
SUM	TOTAL ELAPSED TIME IN THE SIMULATION (T);
SUMP	TOTAL ELAPSED TIME IN THE PUMPING PERIOD (T);
SURI	HYDRAULIC HEAD AT THE START OF THE SIMULATION (L);
SY	SPECIFIC YIELD;
T	TRANSMISSIVITY (L**2/T);
TC	HARMONIC AVERAGE OF T/DELY @ I+1/2,J (L/T);

Definition of program variables—Continued

```
TEMP      VECTOR FOR TEMPORARY STORAGE OF HYDRAULIC HEAD (L);
TEST      = 0 CLOSURE CRITERION SATISFIED;
          = 1 CLOSURE CRITERION NOT SATISFIED;
TEST2     MAXIMUM CHANGE IN HEAD FOR THE TIME STEP (L);
TEST3     VECTOR CONTAINING THE SUM OF THE ABSOLUTE VALUES
          OF HEAD CHANGES FOR EACH ITERATION (L);
TL        TRANSIENT PART OF LEAKAGE COEFFICIENT (1/T);
TMAX      NUMBER OF DAYS IN THE PUMPING PERIOD (T);
TMIN      MINIMUM VALUE OF DIMENSIONLESS TIME FOR THE CURRENT
          PUMPING PERIOD;
TOP       ELEVATION OF THE TOP OF THE AQUIFER (L);
TR        HARMONIC AVERAGE OF T/DELX @ I,J+1/2 (L/T);
TT        MAXIMUM VALUE OF DIMENSIONLESS TIME FOR THE CURRENT
          PUMPING PERIOD;
U         = 0 EXPLICIT TREATMENT OF TRANSIENT LEAKAGE;
          = 1 IMPLICIT TREATMENT OF TRANSIENT LEAKAGE;
U         INDICATES DEFINE FILE RECORD LENGTH SPECIFICATION IN WORDS;
V         ARRAY USED IN SIP ALGORITHM;
VF4       VARIABLE FORMAT FOR PRINTING HEAD AND DRAWDOWN;
WATER     CONTAINS CHARACTER STRING FOR WATER TABLE OPTION;
WELL      WELL DISCHARGE (L**3/T);
WR        WELL RADIUS (L);
XI        ARRAY CONTAINING INCREMENTAL HEAD VALUES IN SIP SOLUTION (L);
Y         VECTOR CONTAINING ARRAY STORAGE;
YDIM      LENGTH OF AQUIFER IN Y DIRECTION (L).

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);
XNET      NET LEAKAGE RATE FOR A CELL (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;
```

Definition of variables in the PRNTAI subroutine—Continued

FACT1	FACTOR FOR ADJUSTING VALUE OF DRAWDOWN PRINTED;
FACT2	FACTOR FOR ADJUSTING VALUE OF HEAD PRINTED;
K	ADJUSTED VALUE OF DRAWDOWN OR HEAD;
MESUR	NAME OF MAP LENGTH UNIT;
N	INDEX FOR SYMBOLS;
NA	INDICES FOR LOCATING X LABEL;
NC	NUMBER OF BLANKS BEFORE GRAPH;
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;
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;
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.

Attachment VII

Program Listing

```
C     ******************************************************************
C                          FINITE-DIFFERENCE MODEL
C                                    FOR
C                     SIMULATION OF GROUND-WATER FLOW
C                            IN TWO DIMENSIONS
C
C             BY P. C. TRESCOTT, G. F. PINDER AND S. P. LARSON
C                          U. S. GEOLOGICAL SURVEY
C                              SEPTEMBER, 1975
C     ******************************************************************
C     MAIN PROGRAM TO DIMENSION DIGITAL MODEL AND CONTROL SEQUENCE
C     OF COMPUTATIONS
C     ------------------------------------------------------------------
C     SPECIFICATIONS:
      REAL *4KEEP,M,HEADNG(32)
      REAL *8PHI,G,BE,TEMP,Z,YY
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
C
      DIMENSION Y(70000), L(37), IFMT1(9), IFMT2(9), IFMT3(9), NAME(99),
     1 YY(1)
      EQUIVALENCE (YY(1),Y(1))
C
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LE
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DI
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2
C
      DATA IFMT1/4H(1H0,4H,I5,,4H10E1,4H1.3/,4H(1H ,4H,5X,,4H10E1,4H1.3)
     1,4H)   /
      DATA IFMT2/4H('0',4H,I2,,4H2X,2,4H0F6.,4H1/(5,4HX,20,4HF6.1,4H))
     1,4H    /
      DATA IFMT3/4H(1H0,4H,I5,,4H14F9,4H.5/(,4H1H ,,4H5X,1,4H4F9.,4H5))
     1,4H    /
      DATA NAME/2*4H    ,4H STO,4HRAGE,4H COE,4HFFIC,4HIENT,4*4H    ,4H 
     1  T,4HRANS,4HMISS,4HIVIT,4HY   ,2*4H    ,4H   A,4HQUIF,4HER H,4HYC
     2RA,4HULIC,4H CON,4HDUCT,4HIVIT,4HY   ,4H    ,4H   A,4HQUIF,4HER B,
     34HASE ,4HELEV,4HATIO,4HN   ,3*4H    ,4H   S,4HPECI,4HFIC ,4HYIEL,4
     4HD   ,4*4H    ,4HAQUI,4HFER ,4HTOP ,4HELEV,4HATIO,4HN   ,4H    ,4H
     5CONF,4HININ,4HG BE,4HD HY,4HDRAU,4HLIC ,4HCOND,4HUCTI,4HVITY,3*4H 
     6   ,4H RIV,4HER H,4HEAD ,4*4H    ,4H   C,4HONFI,4HNING,4H BED,4H T
     7HI,4HCKNE,4HSS  ,2*4H    ,4H   L,4HAND ,4HSURF,4HACE ,4HELEV,4HATI
     8O,4HN   ,3*4H    ,4H ARE,4HAL R,4HECHA,4HRGE ,4HRATE,2*4H    /
C
      DEFINE FILE 2(14,2624,U,KKK)
C     ..................................................................
C
C     ---READ TITLE,PROGRAM OPTIONS AND PROGRAM SIZE---
   10 READ (R,370) HEADNG
      WRITE (P,360) HEADNG
      READ (R,380) WATER,LEAK,CONVRT,EVAP,RECH,NUMS,CHCK,PNCH,IDK1,IDK2,
     1NUM,HEAD
      WRITE (P,390) WATER,LEAK,CONVRT,EVAP,RECH,NUMS,CHCK,PNCH,IDK1,IDK2
     1,NUM,HEAD
```

Program listing—Continued

```
      IF (NUMS.EQ.CHK(11).OR.NUMS.EQ.CHK(12).OR.NUMS.EQ.CHK(13)) GO TO 2
     10
      WRITE (P,350)
      STOP
   20 READ (R,320) DIML,DIMW,NW,ITMAX
      WRITE (P,340) DIML,DIMW,NW,ITMAX
C
C     ---COMPUTE DIMENSIONS FOR ARRAYS---
      IZ=DIML
      JZ=DIMW
      IH=MAX0(1,NW)
      IMAX=MAX0(DIML,DIMW)
      ISIZ=DIML*DIMW
      ISUM=2*ISIZ+1
      IMX1=ITMAX+1
      L(1)=1
      DO 30 I=2,4
      L(I)=ISUM
   30 ISUM=ISUM+2*IMAX
      DO 40 I=5,16
      L(I)=ISUM
   40 ISUM=ISUM+ISIZ
      IF (WATER.NE.CHK(2)) GO TO 60
      DO 50 I=17,19
      L(I)=ISUM
   50 ISUM=ISUM+ISIZ
      IP=DIML
      JP=DIMW
      GO TO 80
   60 DO 70 I=17,19
      L(I)=ISUM
   70 ISUM=ISUM+1
      IP=1
      JP=1
   80 IF (LEAK.NE.CHK(9)) GO TO 100
      DO 90 I=20,22
      L(I)=ISUM
   90 ISUM=ISUM+ISIZ
      IR=DIML
      JR=DIMW
      GO TO 120
  100 DO 110 I=20,22
      L(I)=ISUM
  110 ISUM=ISUM+1
      IR=1
      JR=1
  120 IF (CONVRT.NE.CHK(7)) GO TO 130
      L(23)=ISUM
      ISUM=ISUM+ISIZ
      IC=DIML
      JC=DIMW
      GO TO 140
  130 L(23)=ISUM
      ISUM=ISUM+1
      IC=1
      JC=1
  140 IF (EVAP.NE.CHK(6)) GO TO 150
      L(24)=ISUM
      ISUM=ISUM+ISIZ
      IL=DIML
      JL=DIMW
      GO TO 160
```

Program listing—Continued

```
  150 L(24)=ISUM                                                        MAN1200
      ISUM=ISUM+1                                                       MAN1210
      IL=1                                                              MAN1220
      JL=1                                                              MAN1230
  160 IF (NUMS.NE.CHK(11)) GO TO 180                                    MAN1240
      DO 170 I=25,28                                                    MAN1250
      L(I)=ISUM                                                         MAN1260
  170 ISUM=ISUM+ISIZ                                                    MAN1270
      IS=DIML                                                           MAN1280
      JS=DIMW                                                           MAN1290
      GO TO 200                                                         MAN1300
  180 DO 190 I=25,28                                                    MAN1310
      L(I)=ISUM                                                         MAN1320
  190 ISUM=ISUM+1                                                       MAN1330
      IS=1                                                              MAN1340
      JS=1                                                              MAN1350
  200 DO 210 I=29,31                                                    MAN1360
      L(I)=ISUM                                                         MAN1370
  210 ISUM=ISUM+DIMW                                                    MAN1380
      DO 220 I=32,33                                                    MAN1390
      L(I)=ISUM                                                         MAN1400
  220 ISUM=ISUM+DIML                                                    MAN1410
      L(34)=ISUM                                                        MAN1420
      ISUM=ISUM+IH                                                      MAN1430
      L(35)=ISUM                                                        MAN1440
      ISUM=ISUM+2*IH                                                    MAN1450
      IF (MOD(ISUM,2).EQ.0) ISUM=ISUM+1                                 MAN1460
      CONTINUE                                                          MAN1470
  230 L(36)=ISUM                                                        MAN1480
      ISUM=ISUM+2*IMAX                                                  MAN1490
      L(37)=ISUM                                                        MAN1500
      ISUM=ISUM+IMX1                                                    MAN1510
      WRITE (P,330) ISUM                                                MAN1520
C                                                                       MAN1530
C     ---PASS INTIIAL ADDRESSES OF ARRAYS TO SUBROUTINES---             MAN1540
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L( MAN1550
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))MAN1560
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(MAN1570
     3L(34)),Y(L(35)))                                                  MAN1580
      CALL STEP(Y(L(1)),Y(L(5)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(14)),Y(L(17)MAN1590
     1),Y(L(18)),Y(L(23)),Y(L(29)),Y(L(30)),Y(L(32)),Y(L(34)),Y(L(35)),YMAN1600
     2(L(37)))                                                          MAN1610
      IF (NUMS.EQ.CHK(11)) CALL SOLVE1(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),YMAN1620
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))MAN1630
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(MAN1640
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2MAN1650
     43)),Y(L(20)),Y(L(22)),Y(L(21)))                                   MAN1660
      IF (NUMS.EQ.CHK(12)) CALL SOLVE2(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),YMAN1670
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))MAN1680
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(MAN1690
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2MAN1700
     43)),Y(L(20)),Y(L(22)),Y(L(21)))                                   MAN1710
      IF (NUMS.EQ.CHK(13)) CALL SOLVE3(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),YMAN1720
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))MAN1730
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(MAN1740
     3L(32)),Y(L(33)),Y(L(36)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(1MAN1750
     49)),Y(L(23)),Y(L(20)),Y(L(22)),Y(L(21)))                          MAN1760
      CALL COEF(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10))MAN1770
     1,Y(L(11)),Y(L(12)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(MAN1780
     2L(19)),Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(3MAN1790
     32)))                                                              MAN1800
```

Program listing—Continued

```
      CALL CHECKI(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(10)),Y(L(1MAN1810
     11)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(17)),Y(L(18)),Y(L(19))MAN1820
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)))  MAN1830
      CALL PRNTAI(Y(L(1)),Y(L(8)),Y(L(9)),Y(L(12)),Y(L(14)),Y(L(29)),Y(LMAN1840
     1(32)))                                                            MAN1850
C     ..................................................................MAN1860
C                                                                       MAN1870
C     ---START COMPUTATIONS---                                          MAN1880
C     ************************                                          MAN1890
C     ---READ AND WRITE DATA FOR GROUPS II AND III---                   MAN1900
      CALL DATAIN                                                       MAN1910
      CALL ARRAY(Y(L(12)),IFMT3,NAME(1),2)                              MAN1920
      IF (WATER.EQ.CHK(2)) GO TO 240                                    MAN1930
      CALL ARRAY(Y(L(9)),IFMT3,NAME(10),3)                              MAN1940
      GO TO 250                                                         MAN1950
  240 CALL ARRAY(Y(L(17)),IFMT1,NAME(19),4)                             MAN1960
      CALL ARRAY(Y(L(18)),IFMT2,NAME(28),5)                             MAN1970
      CALL ARRAY(Y(L(19)),IFMT3,NAME(37),6)                             MAN1980
  250 IF (CONVRT.EQ.CHK(7)) CALL ARRAY(Y(L(23)),IFMT2,NAME(46),7)       MAN1990
      IF (LEAK.NE.CHK(9)) GO TO 260                                     MAN2000
      CALL ARRAY(Y(L(20)),IFMT1,NAME(55),8)                             MAN2010
      CALL ARRAY(Y(L(21)),IFMT2,NAME(64),9)                             MAN2020
      CALL ARRAY(Y(L(22)),IFMT2,NAME(73),10)                            MAN2030
  260 IF (EVAP.EQ.CHK(6)) CALL ARRAY(Y(L(24)),IFMT2,NAME(82),11)        MAN2040
      IF (RECH.EQ.CHK(10)) CALL ARRAY(Y(L(13)),IFMT1,NAME(91),12)       MAN2050
      CALL MDAT                                                         MAN2060
C                                                                       MAN2070
C     ---INITIALIZE TRANSMISSIVITY VALUES IN WATER TABLE PROBLEM---     MAN2080
      KT=0                                                              MAN2090
      IF (WATER.EQ.CHK(2)) CALL TRANS                                   MAN2100
C                                                                       MAN2110
C     ---COMPUTE ITERATION PARAMETERS---                                MAN2120
      IF (NUMS.EQ.CHK(11)) CALL ITER1                                   MAN2130
      IF (NUMS.EQ.CHK(12)) CALL ITER2                                   MAN2140
      IF (NUMS.EQ.CHK(13)) CALL ITER3                                   MAN2150
C                                                                       MAN2160
C     ---INITIALIZE PARAMETERS FOR ALPHAMERIC MAP---                    MAN2170
      IF (CONTR.EQ.CHK(3)) CALL MAP                                     MAN2180
C                                                                       MAN2190
C     ---COMPUTE T COEFFICIENTS FOR ARTESIAN PROBLEM---                 MAN2200
      IF (WATER.NE.CHK(2)) CALL TCOF                                    MAN2210
C                                                                       MAN2220
C     ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-MAN2230
  270 CALL NEWPER                                                       MAN2240
C                                                                       MAN2250
      KT=0                                                              MAN2260
      IFINAL=0                                                          MAN2270
      IERR=0                                                            MAN2280
C                                                                       MAN2290
C     ---START NEW TIME STEP COMPUTATIONS---                            MAN2300
  280 CALL NEWSTP                                                       MAN2310
C                                                                       MAN2320
C     ---COMPUTE TRANSIENT PART OF LEAKAGE TERM---                      MAN2330
      IF (LEAK.EQ.CHK(9).AND.SS.NE.0.) CALL CLAY                        MAN2340
C                                                                       MAN2350
C     ---ENTER APPROPIATE SOLUTION ROUTINE AND COMPUTE SOLUTION---      MAN2360
      IF (NUMS.EQ.CHK(11)) CALL NEWITA                                  MAN2370
      IF (NUMS.EQ.CHK(12)) CALL NEWITB                                  MAN2380
      IF (NUMS.EQ.CHK(13)) CALL NEWITC                                  MAN2390
C                                                                       MAN2400
C     ---CHECK FOR STEADY STATE AND PRINT OUTPUT AT DESIGNATED          MAN2410
```

Program listing—Continued

```
C     TIME STEPS---
      CALL STEADY
C
C     ---LAST TIME STEP IN PUMPING PERIOD ?---
      IF (IFINAL.NE.1) GO TO 280
C
C     ---CHECK FOR NEW PUMPING PERIOD---
      IF (KP.LT.NPER) GO TO 270
C
C     ---DISK OUTPUT IF DESIRED---
      IF (IDK2.NE.CHK(15)) GO TO 290
      CALL DISK
C
C     ---PUNCHED OUTPUT IF DESIRED---
  290 IF (PNCH.NE.CHK(1)) GO TO 300
      CALL PUNCH
C
C     ---CHECK FOR NEW PROBLEM---
  300 READ (R,320,END=310) NEXT
      IF (NEXT.EQ.0) GO TO 10
  310 STOP
C     ..................................................................
C
C     ---FORMATS---
C     ------------------------------------------------------------------
C
C
  320 FORMAT (4I10)
  330 FORMAT ('0',54X,'WORDS OF Y VECTOR USED =',I7)
  340 FORMAT ('0',62X,'NUMBER OF ROWS =',I5/60X,'NUMBER OF COLUMNS =',I5
     1/9X,'NUMBER OF WELLS FOR WHICH DRAWDOWN IS COMPUTED AT A SPECIFIED
     2 RADIUS =',I5,/,39X,'MAXIMUM PERMITTED NUMBER OF ITERATIONS =',I5)
  350 FORMAT ('-',36X,'NO EQUATION SOLVING SCHEME SPECIFIED, EXECUTION T
     1ERMINATED'/37X,58('*'))
  360 FORMAT ('1',60X,'U. S. G. S.'//55X,'FINITE-DIFFERENCE MODEL'/65X,'
     1FOR'/51X,'SIMULATION OF GROUND-WATER FLOW'//60X,'JANUARY, 1975'//1
     233('*')/'0',32A4//133('*'))
  370 FORMAT (20A4)
  380 FORMAT (16(A4,1X))
  390 FORMAT ('-SIMULATION OPTIONS:  ',13(A4,4X))
      END

      SUBROUTINE DATAI(PHI,STRT,SURI,T,TR,TC,S,QRE,WELL,TL,SL,PERM,BOTTO
     1M,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,WR,NWR)
C     ------------------------------------------------------------------
C     READ AND WRITE INPUT DATA
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI,DBLE,XLABEL,YLABEL,TITLE,XN1,MESUR
      REAL *4M
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
C
      DIMENSION PHI(IZ,JZ), STRT(IZ,JZ), SURI(IZ,JZ), T(IZ,JZ), TR(IZ,JZ
     1), TC(IZ,JZ), S(IZ,JZ), QRE(IZ,JZ), WELL(IZ,JZ), TL(IZ,JZ), SL(IZ,
     2JZ), PERM(IP,JP), BOTTOM(IP,JP), SY(IP,JP), RATE(IR,JR), RIVER(IR,
     3JR), M(IR,JR), TOP(IC,JC), GRND(IL,JL), DELX(JZ), DELY(IZ), WR(IH)
     4, NWR(IH,2), A(IZ,JZ), IN(9), IFMT(9)
C
```

Program listing—Continued

```
      COMMON /SARRAY/ VF4(11),CHK(15)                                   DAT 190
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LEDAT 200
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,DAT 210
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,DAT 220
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DIDAT 230
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2                                 DAT 240
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT  DAT 250
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(5),XN1,MESUR,PRNT(122),BLANKDAT 260
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),DAT 270
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                          DAT 280
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1  DAT 290
      RETURN                                                            DAT 300
C     ..................................................................DAT 310
C     **********************                                            DAT 320
      ENTRY DATAIN                                                      DAT 330
C     **********************                                            DAT 340
C                                                                       DAT 350
C     ---READ AND WRITE SCALAR PARAMETERS---                            DAT 360
      READ (R,500) CONTR,XSCALE,YSCALE,DINCH,FACT1,FACT2,MESUR          DAT 370
      IF (CONTR.EQ.CHK(3)) WRITE (P,610) XSCALE,YSCALE,MESUR,MESUR,DINCHDAT 380
     1,FACT1,FACT2                                                      DAT 390
      READ (R,490) NPER,KTH,ERR,EROR,SS,QET,ETDIST,LENGTH,HMAX,FACTX,FACDAT 400
     1TY                                                                DAT 410
      IF (ETDIST.LE.0.) ETDIST=1.                                       DAT 420
      WRITE (P,520) NPER,KTH,ERR,EROR,SS,QET,ETDIST,FACTX,FACTY         DAT 430
C                                                                       DAT 440
C     ---READ CUMULATIVE MASS BALANCE PARAMETERS---                     DAT 450
      READ (R,600) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT 460
     1XT,FLXNT                                                          DAT 470
      IF (IDK1.EQ.CHK(14)) GO TO 20                                     DAT 480
      IF (SUM.EQ.0.0) GO TO 40                                          DAT 490
      WRITE (P,480) SUM                                                 DAT 500
C     ..................................................................DAT 510
C                                                                       DAT 520
C     ---HEAD DATA TO CONTINUE PREVIOUS COMPUTATIONS READ HERE---       DAT 530
C     ------FROM CARDS:                                                 DAT 540
      DO 10 I=1,DIML                                                    DAT 550
      READ (R,540) (PHI(I,J),J=1,DIMW)                                  DAT 560
   10 WRITE (P,530) I,(PHI(I,J),J=1,DIMW)                               DAT 570
      GO TO 40                                                          DAT 580
C     ------READ AND WRITE DATA FROM UNIT 4 ON DISK RATHER THAN CARDS:  DAT 590
   20 READ (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT 600
     1XT,FLXNT                                                          DAT 610
      WRITE (P,480) SUM                                                 DAT 620
      DO 30 I=1,DIML                                                    DAT 630
   30 WRITE (P,530) I,(PHI(I,J),J=1,DIMW)                               DAT 640
      REWIND 4                                                          DAT 650
C     ............................... STRT (STARTING HEAD) .............DAT 660
   40 READ (R,490) FACT,IVAR,IPRN,IRECS,IRECD                           DAT 670
      IF (IRECS.EQ.1) READ (2'1) STRT                                   DAT 680
      IF ((IVAR.EQ.1.OR.IRECS.EQ.1).AND.IPRN.NE.1) WRITE (P,470)        DAT 690
      DO 80 I=1,DIML                                                    DAT 700
      IF (IVAR.EQ.1) READ (R,540) (STRT(I,J),J=1,DIMW)                  DAT 710
      DO 70 J=1,DIMW                                                    DAT 720
      IF (IRECS.EQ.1) GO TO 60                                          DAT 730
      IF (IVAR.NE.1) GO TO 50                                           DAT 740
      STRT(I,J)=STRT(I,J)*FACT                                          DAT 750
      GO TO 60                                                          DAT 760
   50 STRT(I,J)=FACT                                                    DAT 770
   60 SURI(I,J)=STRT(I,J)                                               DAT 780
      T(I,J)=0.                                                         DAT 785
```

Program listing—Continued

```
      TL(I,J)=0.
      SL(I,J)=0.
      TR(I,J)=0.
      TC(I,J)=0.
      WELL(I,J)=0.0
      QRE(I,J)=0.
   70 IF (SUM.EQ.0.0.AND.IDK1.NE.CHK(14)) PHI(I,J)=STRT(I,J)
      IF (IVAR.EQ.0.AND.IRECS.EQ.0.OR.IPRN.EQ.1) GO TO 80
      WRITE (P,530) I,(STRT(I,J),J=1,DIMW)
   80 CONTINUE
      IF (IVAR.NE.1.AND.IRECS.NE.1) WRITE (P,420) FACT
      IF (IRECD.EQ.1) WRITE (2'1) STRT
      RETURN
C
C     ---READ REMAINING ARRAYS FROM CARDS OR DISK (AS SPECIFIED IN THE
C        OPTIONS) AND WRITE THEM ON DISK IF SPECIFIED IN THE OPTIONS---
C     **************
      ENTRY ARRAY(A,IFMT,IN,IRN)
C     **************
      READ (R,490) FACT,IVAR,IPRN,IRECS,IRECD
      IK=4*IRECS+2*IVAR+IPRN+1
      GO TO (90,90,110,110,140,140), IK
   90 DO 100 I=1,DIML
      DO 100 J=1,DIMW
  100 A(I,J)=FACT
      WRITE (P,430) IN,FACT
      GO TO 160
  110 IF (IK.EQ.3) WRITE (P,440) IN
      DO 130 I=1,DIML
      READ (R,510) (A(I,J),J=1,DIMW)
      DO 120 J=1,DIMW
  120 A(I,J)=A(I,J)*FACT
  130 IF (IK.EQ.3) WRITE (P,IFMT) I,(A(I,J),J=1,DIMW)
      GO TO 160
  140 READ (2'IRN) A
      IF (IK.EQ.6) GO TO 160
      WRITE (P,440) IN
      DO 150 I=1,DIML
  150 WRITE (P,IFMT) I,(A(I,J),J=1,DIMW)
  160 IF (IRECD.EQ.1) WRITE (2'IRN) A
      RETURN
C
C     ---INSERT ZERO VALUES IN THE T OR PERM MATRIX AROUND THE
C        BORDER OF THE MODEL---
C     **************
      ENTRY MDAT
C     **************
      DO 180 I=1,DIML
      DO 180 J=1,DIMW
      IF (WATER.EQ.CHK(2)) GO TO 170
      IF (I.EQ.1.OR.I.EQ.DIML.OR.J.EQ.1.OR.J.EQ.DIMW) T(I,J)=0.
      GO TO 180
  170 IF (I.EQ.1.OR.I.EQ.DIML.OR.J.EQ.1.OR.J.EQ.DIMW) PERM(I,J)=0.
  180 CONTINUE
C     ............................... DELX,DELY .......................
      READ (R,490) FACT,IVAR,IPRN,IRECS,IRECD
      IF (IRECS.EQ.1) GO TO 210
      IF (IVAR.EQ.1) READ (R,490) DELX
      DO 200 J=1,DIMW
      IF (IVAR.NE.1) GO TO 190
      DELX(J)=DELX(J)*FACT
```

Program listing—Continued

```
      GO TO 200                                                         DAT1400
  190 DELX(J)=FACT                                                      DAT1410
  200 CONTINUE                                                          DAT1420
      GO TO 220                                                         DAT1430
  210 READ (2'13) DELX                                                  DAT1440
  220 IF (IRECD.EQ.1) WRITE (2'13) DELX                                 DAT1450
      IF (IVAR.EQ.1.OR.IRECS.EQ.1.AND.IPRN.NE.1) WRITE (P,550) DELX     DAT1460
      IF (IVAR.NE.1.AND.IRECS.NE.1) WRITE (P,450) FACT                  DAT1470
      READ (R,490) FACT,IVAR,IPRN,IRECS,IRECD                           DAT1480
      IF (IRECS.EQ.1) GO TO 250                                         DAT1490
      IF (IVAR.EQ.1) READ (R,490) DELY                                  DAT1500
      DO 240 I=1,DIML                                                   DAT1510
      IF (IVAR.NE.1) GO TO 230                                          DAT1520
      DELY(I)=DELY(I)*FACT                                              DAT1530
      GO TO 240                                                         DAT1540
  230 DELY(I)=FACT                                                      DAT1550
  240 CONTINUE                                                          DAT1560
      GO TO 260                                                         DAT1570
  250 READ (2'14) DELY                                                  DAT1580
  260 IF (IRECD.EQ.1) WRITE (2'14) DELY                                 DAT1590
      IF (IVAR.EQ.1.OR.IRECS.EQ.1.AND.IPRN.NE.1) WRITE (P,560) DELY     DAT1600
      IF (IVAR.NE.1.AND.IRECS.NE.1) WRITE (P,460) FACT                  DAT1610
C                                                                       DAT1620
C     ---INITIALIZE VARIABLES---                                        DAT1630
      JNO1=DIMW-1                                                       DAT1640
      INO1=DIML-1                                                       DAT1650
      IF (LEAK.NE.CHK(9).OR.SS.NE.0.) GO TO 280                         DAT1660
      DO 270 I=2,INO1                                                   DAT1670
      DO 270 J=2,JNO1                                                   DAT1680
      IF (M(I,J).EQ.0.) GO TO 270                                       DAT1690
      TL(I,J)=RATE(I,J)/M(I,J)                                          DAT1700
  270 CONTINUE                                                          DAT1710
  280 ETQB=0.0                                                          DAT1720
      ETQD=0.0                                                          DAT1730
      SUBS=0.0                                                          DAT1740
      U=1.0                                                             DAT1750
      TT=0.0                                                            DAT1760
      IM=MIN0(6*DIMW+4,124)                                             DAT1770
      IM=(132-IM)/2                                                     DAT1780
      VF4(3)=DIGIT(IM)                                                  DAT1790
      VF4(8)=DIGIT(IM+5)                                                DAT1800
      WIDTH=0.                                                          DAT1810
      DO 290 J=2,JNO1                                                   DAT1820
  290 WIDTH=WIDTH+DELX(J)                                               DAT1830
      YDIM=0.                                                           DAT1840
      DO 300 I=2,INO1                                                   DAT1850
  300 YDIM=YDIM+DELY(I)                                                 DAT1860
      RETURN                                                            DAT1870
C     ..................................................................DAT1880
C                                                                       DAT1890
C     ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-DAT1900
C     ************************                                          DAT1910
      ENTRY NEWPER                                                      DAT1920
C     ************************                                          DAT1930
C                                                                       DAT1940
      READ (R,490) KP,KPM1,NWEL,TMAX,NUMT,CDLT,DELT                     DAT1950
C                                                                       DAT1960
C     ---COMPUTE ACTUAL DELT AND NUMT---                                DAT1970
      DT=DELT/24.                                                       DAT1980
      TM=0.0                                                            DAT1990
      DO 310 I=1,NUMT                                                   DAT2000
```

Program listing—Continued

```
      DT=CDLT*DT                                                        DAT2010
      TM=TM+DT                                                          DAT2020
      IF (TM.GE.TMAX) GO TO 320                                         DAT2030
  310 CONTINUE                                                          DAT2040
      GO TO 330                                                         DAT2050
  320 DELT=TMAX/TM*DELT                                                 DAT2060
      NUMT=I                                                            DAT2070
  330 WRITE (P,570) KP,TMAX,NUMT,DELT,CDLT                              DAT2080
      DELT=DELT*3600.                                                   DAT2090
      TMAX=TMAX*86400.                                                  DAT2100
C                                                                       DAT2110
C     ---INITIALIZE SUMP, STRT, SL, WELL AND WR---                      DAT2120
      WRITE (P,580) NWEL                                                DAT2130
      IF (KP.GT.KPM1) SUMP=0.                                           DAT2140
      DO 350 I=1,DIML                                                   DAT2150
      DO 350 J=1,DIMW                                                   DAT2160
      IF (KP.EQ.KPM1) GO TO 340                                         DAT2170
      STRT(I,J)=PHI(I,J)                                                DAT2180
  340 IF (LEAK.NE.CHK(9)) GO TO 350                                     DAT2190
      IF (M(I,J).EQ.0.) GO TO 350                                       DAT2200
      SL(I,J)=RATE(I,J)/M(I,J)*(RIVER(I,J)-STRT(I,J))                   DAT2210
  350 WELL(I,J)=0.                                                      DAT2220
      IF (NW.EQ.0) GO TO 370                                            DAT2230
      DO 360 I=1,NW                                                     DAT2240
  360 WR(I)=0.                                                          DAT2250
  370 IF (NWEL.EQ.0) GO TO 410                                          DAT2260
C                                                                       DAT2270
C     ---READ AND WRITE WELL PUMPING RATES AND WELL RADII---            DAT2280
      KW=0                                                              DAT2290
      DO 400 II=1,NWEL                                                  DAT2300
      READ (R,490) I,J,WELL(I,J),RADIUS                                 DAT2310
      IF (RADIUS.EQ.0.) GO TO 380                                       DAT2320
      KW=KW+1                                                           DAT2330
      IF (KW.GT.NW) GO TO 380                                           DAT2340
      NWR(KW,1)=I                                                       DAT2350
      NWR(KW,2)=J                                                       DAT2360
      WR(KW)=RADIUS                                                     DAT2370
      WRITE (P,590) I,J,WELL(I,J),WR(KW)                                DAT2380
      GO TO 390                                                         DAT2390
  380 WRITE (P,590) I,J,WELL(I,J)                                       DAT2400
  390 WELL(I,J)=WELL(I,J)/(DELX(J)*DELY(I))                             DAT2410
  400 CONTINUE                                                          DAT2420
  410 RETURN                                                            DAT2430
C     ..................................................................DAT2440
C                                                                       DAT2450
C     FORMATS:                                                          DAT2460
C                                                                       DAT2470
C     ------------------------------------------------------------------DAT2480
C                                                                       DAT2490
C                                                                       DAT2500
  420 FORMAT ('0',63X,'STARTING HEAD =',G15.7)                          DAT2510
  430 FORMAT ('0',41X,9A4,'=',G15.7)                                    DAT2520
  440 FORMAT ('1',49X,9A4,/,65X,'MATRIX',/,50X,36('-'))                 DAT2530
  450 FORMAT ('0',72X,'DELX =',G15.7)                                   DAT2540
  460 FORMAT ('0',72X,'DELY =',G15.7)                                   DAT2550
  470 FORMAT ('1',60X,'STARTING HEAD MATRIX'/61X,20('-'))               DAT2560
  480 FORMAT ('1',40X,' CONTINUATION - HEAD AFTER ',G20.7,' SEC PUMPING DAT2570
     1'/42X,58('-'))                                                    DAT2580
  490 FORMAT (8G10.0)                                                   DAT2590
  500 FORMAT (A4,6X,5G10.0,A8)                                          DAT2600
  510 FORMAT (20F4.0)                                                   DAT2610
```

Program listing—Continued

```
  520 FORMAT ('0',51X,'NUMBER OF PUMPING PERIODS =',I5/49X,'TIME STEPS BDAT2620
     1ETWEEN PRINTOUTS =',I5//51X,'ERROR CRITERIA FOR CLOSURE =',G15.7/4DAT2630
     21X,'          STEADY STATE ERROR CRITERIA =',G15.7//44X,'SPECIFIC SDAT2640
     3TORAGE OF CONFINING BED =',G15.7/54X,'EVAPOTRANSPIRATION RATE =',GDAT2650
     415.7/56X,'EFFECTIVE DEPTH OF ET =',G15.7//22X,'MULTIPLICATION FACTDAT2660
     5OR FOR TRANSMISSIVITY IN X DIRECTION =',G15.7/63X,'IN Y DIRECTION DAT2670
     6=',G15.7)                                                         DAT2680
  530 FORMAT ('0',I2,2X,20F6.1/(5X,20F6.1))                             DAT2690
  540 FORMAT (8F10.4)                                                   DAT2700
  550 FORMAT (1H1,46X,40HGRID SPACING IN PROTOTYPE IN X DIRECTION/47X,40DAT2710
     1('-')//('0',12F10.0))                                             DAT2720
  560 FORMAT (1H-,46X,40HGRID SPACING IN PROTOTYPE IN Y DIRECTION/47X,40DAT2730
     1('-')//('0',12F10.0))                                             DAT2740
  570 FORMAT ('-',50X,'PUMPING PERIOD NO.',I4,':',F10.2,' DAYS'/51X,38('DAT2750
     1-')//53X,'NUMBER OF TIME STEPS=',I6//59X,'DELT IN HOURS =',F10.3//DAT2760
     253X,'MULTIPLIER FOR DELT =',F10.3)                                DAT2770
  580 FORMAT ('-',63X,I4,' WELLS'/65X,9('-')//50X,'I',9X,'J    PUMPING RDAT2780
     1ATE  WELL RADIUS'/)                                               DAT2790
  590 FORMAT (41X,2I10,2F13.2)                                          DAT2800
  600 FORMAT (4G20.10)                                                  DAT2810
  610 FORMAT ('0',30X,'ON ALPHAMERIC MAP:'/40X,'MULTIPLICATION FACTOR FODAT2820
     1R X DIMENSION =',G15.7/40X,'MULTIPLICATION FACTOR FOR Y DIMENSION DAT2830
     2=',G15.7/55X,'MAP SCALE IN UNITS OF  ',A11/50X,'NUMBER OF ',A8,' PDAT2840
     3ER INCH =',G15.7/43X,'MULTIPLICATION FACTOR FOR DRAWDOWN =',G15.7/DAT2850
     447X,'MULTIPLICATION FACTOR FOR HEAD =',G15.7)                     DAT2860
      END                                                               DAT2870-


      SUBROUTINE STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDNSTP  10
     1,DELY,WR,NWR,TEST3)                                               STP  20
C     ------------------------------------------------------------------STP  30
C     INITIALIZE DATA FOR TIME STEP, CHECK FOR STEADY STATE,            STP  40
C     PRINT AND  PUNCH RESULTS                                          STP  50
C     ------------------------------------------------------------------STP  60
C                                                                       STP  70
C     SPECIFICATIONS:                                                   STP  80
      REAL *8PHI,DBLE,DABS,TEST2,DMAX1,XLABEL,YLABEL,XN1,MESUR,TITLE    STP  90
      REAL *4MINS,M,KEEP                                                STP 100
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,STP 110
     1CONTR,LEAK,RECH,SIP,ADI                                           STP 120
C                                                                       STP 130
      DIMENSION PHI(IZ,JZ), KEEP(IZ,JZ), STRT(IZ,JZ), SURI(IZ,JZ), T(IZ,STP 140
     1JZ), BOTTOM(IP,JP), WELL(IZ,JZ), PERM(IP,JP), TOP(IC,JC), DELX(JZ)STP 150
     2, DDN(JZ), DELY(IZ), WR(IH), NWR(IH,2), ITTO(200), TEST3(IMX1)    STP 160
C                                                                       STP 170
      COMMON /SARRAY/ VF4(11),CHK(15)                                   STP 180
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LESTP 190
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,STP 200
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,STP 210
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DISTP 220
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2                                 STP 230
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT  STP 240
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1  STP 250
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(5),XN1,MESUR,PRNT(122),BLANKSTP 260
     1(60),DIGIT(122),VF1(6),VF2(6),VF3(7),XSCALE,DINCH,SYM(17),XN(100),STP 270
     2YN(13),NA(4),N1,N2,N3,YSCALE,FACT1,FACT2                          STP 280
C                                                                       STP 290
      DATA PIE/3.141593/,YYY/Z00000000/                                 STP 300
      RETURN                                                            STP 310
C     ..................................................................STP 320
C                                                                       STP 330
```

Program listing—Continued

```
C     ---START A NEW TIME STEP---                                       STP 340
C     ********************                                              STP 350
      ENTRY NEWSTP                                                      STP 360
C     ********************                                              STP 370
      KT=KT+1                                                           STP 380
      KOUNT=0                                                           STP 390
      DO 10 I=1,DIML                                                    STP 400
      DO 10 J=1,DIMW                                                    STP 410
   10 KEEP(I,J)=PHI(I,J)                                                STP 420
      DELT=CDLT*DELT                                                    STP 430
      SUM=SUM+DELT                                                      STP 440
      SUMP=SUMP+DELT                                                    STP 450
      DAYSP=SUMP/86400.                                                 STP 460
      YRSP=DAYSP/365.                                                   STP 470
      HRS=SUM/3600.                                                     STP 480
      MINS=HRS*60.                                                      STP 490
      DAYS=HRS/24.                                                      STP 500
      YRS=DAYS/365.                                                     STP 510
      RETURN                                                            STP 520
C     ..................................................................STP 530
C                                                                       STP 540
C     ---CHECK FOR STEADY STATE---                                      STP 550
C     *********************                                             STP 560
      ENTRY STEADY                                                      STP 570
C     *********************                                             STP 580
      TEST2=0.                                                          STP 590
      DO 20 I=2,INO1                                                    STP 600
      DO 20 J=2,JNO1                                                    STP 610
   20 TEST2=DMAX1(TEST2,DABS(DBLE(KEEP(I,J))-PHI(I,J)))                 STP 620
      IF (TEST2.GE.EROR) GO TO 30                                       STP 630
      WRITE (P,330) KT                                                  STP 640
      IFINAL=1                                                          STP 650
      GO TO 40                                                          STP 660
   30 IF (KT.EQ.NUMT) IFINAL=1                                          STP 670
C                                                                       STP 680
C     ---ENTRY FOR TERMINATING COMPUTATIONS IF MAXIMUM ITERATIONS       STP 690
C     EXCEEDED---                                                       STP 700
C     *********************                                             STP 710
      ENTRY TERM1                                                       STP 720
C     *********************                                             STP 730
   40 IF (KT.GT.200) WRITE (P,400)                                      STP 740
      ITTO(KT)=KOUNT                                                    STP 750
      IF (KOUNT.LE.ITMAX) GO TO 80                                      STP 760
      IERR=2                                                            STP 770
      KOUNT=KOUNT-1                                                     STP 780
      ITTO(KT)=KOUNT                                                    STP 790
      IF (KT.EQ.1) GO TO 60                                             STP 800
C                                                                       STP 810
C     ---WRITE ON DISK OR PUNCH CARDS AS SPECIFIED IN THE OPTIONS---    STP 820
      XXX=SUM-DELT                                                      STP 830
      IF (IDK2.EQ.CHK(15)) WRITE (4) ((KEEP(I,J),YYY,I=1,DIML),J=1,DIMW)STP 840
     1,XXX,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLXT,FLXNT    STP 850
      IF (PNCH.NE.CHK(1)) GO TO 80                                      STP 860
      WRITE (PU,360) XXX,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETSTP 870
     1FLXT,FLXNT                                                        STP 880
      DO 50 I=1,DIML                                                    STP 890
   50 WRITE (PU,350) (KEEP(I,J),J=1,DIMW)                               STP 900
      GO TO 80                                                          STP 910
   60 IF (IDK2.EQ.CHK(15)) WRITE (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHSTSTP 920
     1,CHDT,FLUXT,STORT,ETFLXT,FLXNT                                    STP 930
      IF (PNCH.NE.CHK(1)) GO TO 80                                      STP 940
```

Program listing—Continued

```
      WRITE (PU,360) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ET
     1FLXT,FLXNT
      DO 70 I=1,DIML
   70 WRITE (PU,350) (PHI(I,J),J=1,DIMW)
C
   80 IF (CHCK.EQ.CHK(5)) CALL CHECK
      IF (IERR.EQ.2) GO TO 90
C
C     ---PRINT OUTPUT AT DESIGNATED TIME STEPS---
      IF (MOD(KT,KTH).NE.0.AND.IFINAL.NE.1) RETURN
   90 WRITE (P,340) KT,DELT,SUM,MINS,HRS,DAYS,YRS,DAYSP,YRSP
      IF (CHCK.EQ.CHK(5)) CALL CWRITE
      IF (TT.NE.0.) WRITE (P,320) TMIN,TT
      KOUNT=KOUNT+1
      WRITE (P,300) (TEST3(J),J=1,KOUNT)
      WRITE (P,290) TEST2
      I3=1
      I5=0
  100 I5=I5+40
      I4=MIN0(KT,I5)
      WRITE (P,390) (I,I=I3,I4)
      WRITE (P,380)
      WRITE (P,370) (ITTO(I),I=I3,I4)
      WRITE (P,380)
      IF (KT.LE.I5) GO TO 110
      I3=I3+40
      GO TO 100
C
C     ---PRINT ALPHAMERIC MAPS---
  110 IF (CONTR.NE.CHK(3)) GO TO 120
      IF (FACT1.NE.0.) CALL PRNTA(1)
      IF (FACT2.NE.0.) CALL PRNTA(2)
  120 IF (HEAD.NE.CHK(8)) GO TO 140
C
C     ---PRINT HEAD MATRIX---
      WRITE (P,310)
      DO 130 I=1,DIML
  130 WRITE (P,VF4) I,(PHI(I,J),J=1,DIMW)
  140 IF (NUM.NE.CHK(4)) GO TO 170
C
C     ---PRINT DRAWDOWN---
      WRITE (P,280)
C     ********************
      ENTRY DRDN
C     ********************
      DO 160 I=1,DIML
      DO 150 J=1,DIMW
  150 DDN(J)=SURI(I,J)-PHI(I,J)
  160 WRITE (P,VF4) I,(DDN(J),J=1,DIMW)
  170 IF (NW.EQ.0.OR.IERR.EQ.1) GO TO 230
C     ..................................................................
C
C     ---COMPUTE APPROXIMATE HEAD FOR PUMPING WELLS---
      WRITE (P,260)
      DO 220 KW=1,NW
      IF (WR(KW).EQ.0.) GO TO 220
      I=NWR(KW,1)
      J=NWR(KW,2)
C
C     COMPUTE EFFECTIVE RADIUS OF WELL IN MODEL---
      RE=(DELX(J)+DELY(I))/9.62
```

Program listing—Continued

```
      IF (WATER.NE.CHK(2)) GO TO 180
      IF (CONVRT.NE.CHK(7)) GO TO 190
      IF (PHI(I,J).LT.TOP(I,J)) GO TO 190
C
C     ---COMPUTATION FOR WELL IN ARTESIAN AQUIFER---
  180 HW=PHI(I,J)+WELL(I,J)*ALOG(RE/WR(KW))/(2.*PIE*T(I,J))*DELX(J)*DELY
     1(I)
      GO TO 210
C
C     ---COMPUTATION FOR WELL IN WATER TABLE AQUIFER
  190 HED=PHI(I,J)-BOTTOM(I,J)
      ARG=HED*HED+WELL(I,J)*ALOG(RE/WR(KW))/(PIE*PERM(I,J))*DELX(J)*DELY
     1(I)
      IF (ARG.GT.0.) GO TO 200
      WRITE (P,270) I,J
      GO TO 220
  200 HW=SQRT(ARG)+BOTTOM(I,J)
C
C     ---COMPUTE DRAWDOWN AT THE WELL AND PRINT RESULTS---
  210 DRAW=SURI(I,J)-HW
      WRITE (P,250) I,J,WR(KW),HW,DRAW
  220 CONTINUE
  230 IF (IERR.NE.2) RETURN
      STOP
C
C     ---DISK OUTPUT---
C     ****************
      ENTRY DISK
C     ****************
      WRITE (4) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETF
     1LXT,FLXNT
      RETURN
C     ..................................................................
C
C     ---PUNCHED OUTPUT---
C     ********************
      ENTRY PUNCH
C     ********************
      WRITE (PU,360) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ET
     1FLXT,FLXNT
      DO 240 I=1,DIML
  240 WRITE (PU,350) (PHI(I,J),J=1,DIMW)
      RETURN
C
C     ..................................................................
C
C     FORMATS:
C
C
C     ------------------------------------------------------------------
C
C
  250 FORMAT (' ',43X,2I5,3F11.2)
  260 FORMAT ('-',50X,'HEAD AND DRAWDOWN IN PUMPING WELLS'/51X,34('-')//
     148X,'I    J   WELL RADIUS    HEAD    DRAWDOWN'//)
  270 FORMAT (' ',43X,2I5,'  WELL IS DRY')
  280 FORMAT (1H1,60X,'DRAWDOWN'/61X,8('-'))
  290 FORMAT ('0MAXIMUM CHANGE IN HEAD FOR THIS TIME STEP =',F10.3/' ',5
     13('-'))
  300 FORMAT ('0MAXIMUM HEAD CHANGE FOR EACH ITERATION:'/' ',39('-')/('0
     1',10F12.4))
```

Program listing—Continued

```
  310 FORMAT ('1',60X,'HEAD MATRIX'/61X,11('-'))                         STP2170
  320 FORMAT ('0DIMENSIONLESS TIME FOR THIS STEP RANGES FROM',G15.7,'   TSTP2180
     10',G15.7)                                                          STP2190
  330 FORMAT ('-*****STEADY STATE AT TIME STEP',I4,'*****')              STP2200
  340 FORMAT (1H1,44X,57('-')/45X,'|',14X,'TIME STEP NUMBER =',I9,14X,'|STP2210
     1'/45X,57('-')//50X,29HSIZE OF TIME STEP IN SECONDS=,F14.2//55X,'TOSTP2220
     2TAL SIMULATION TIME IN SECONDS=',F14.2/80X,8HMINUTES=,F14.2/82X,6HSTP2230
     3HOURS=,F14.2/83X,5HDAYS=,F14.2/82X,'YEARS=',F14.2///45X,'DURATION STP2240
     4OF CURRENT PUMPING PERIOD IN DAYS=',F14.2/82X,'YEARS=',F14.2//)    STP2250
  350 FORMAT (8F10.4)                                                    STP2260
  360 FORMAT (4G20.10)                                                   STP2270
  370 FORMAT ('0ITERATIONS:',40I3)                                       STP2280
  380 FORMAT (' ',10('-'))                                               STP2290
  390 FORMAT ('0TIME STEP :',40I3)                                       STP2300
  400 FORMAT ('0',10('*'),'THE NUMBER OF TIME STEPS EXCEEDS THE DIMENSIOSTP2310
     1N OF THE VECTOR ITTO AND MAY CAUSE UNEXPECTED RESULTS IN ADDITIONASTP2320
     2L'/'0COMPUTATION.  AVOID PROBLEMS BY INCREASING THE DIMENSION OF TSTP2330
     3HE VECTOR ITTO IN STEP',10('*'))                                   STP2340
      END                                                                STP2350-

      SUBROUTINE SOLVE1(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,DSIP  10
     1EL,ETA,V,XI,DELX,BET,DELY,ALF,TEST3,TR,TC,GRND,SY,TOP,RATE,M,RIVERSIP  20
     2)                                                                  SIP  30
C     -------------------------------------------------------------------SIP  40
C       SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE                      SIP  50
C     -------------------------------------------------------------------SIP  60
C                                                                        SIP  70
C     SPECIFICATIONS:                                                    SIP  80
      REAL *8PHI,DBLE,RHOP(20),G,BE,TEMP,DABS,W,TEST2,DMAX1,RHO,B,D,F,H,SIP  90
     1B1,E,CH,GH,BH,DH,EH,FH,HH,ALFA,BETA,GAMA,RES                       SIP 100
      REAL *4KEEP,M                                                      SIP 110
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,SIP 120
     1CONTR,LEAK,RECH,SIP,IORDER(21),ADI                                 SIP 130
C                                                                        SIP 140
      DIMENSION PHI(1), BE(1), G(1), TEMP(1), KEEP(1), PHE(1), STRT(1), SIP 150
     1T(1), S(1), QRE(1), WELL(1), TL(1), SL(1), DEL(1), ETA(1), V(1), XSIP 160
     2I(1), DELX(1), BET(1), DELY(1), ALF(1), TEST3(1), TR(1), TC(1), GRSIP 170
     3ND(1), SY(1), TOP(1), RATE(1), M(1), RIVER(1)                      SIP 180
C                                                                        SIP 190
      COMMON /SARRAY/ VF4(11),CHK(15)                                    SIP 200
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LESIP 210
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,SIP 220
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,SIP 230
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DISIP 240
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2                                  SIP 250
      RETURN                                                             SIP 260
C     ...................................................................SIP 270
C                                                                        SIP 280
C     ---COMPUTE AND PRINT ITERATION PARAMETERS---                       SIP 290
C     ********************                                               SIP 300
      ENTRY ITER1                                                        SIP 310
C     ********************                                               SIP 320
C     ---INITIALIZE ORDER OF ITERATION PARAMETERS (OR REPLACE WITH A     SIP 330
C     READ STATEMENT)---                                                 SIP 340
      DATA IORDER/1,2,3,4,5,1,2,3,4,5,11*1/                              SIP 350
      I2=INO1-1                                                          SIP 360
      J2=JNO1-1                                                          SIP 370
      L2=LENGTH/2                                                        SIP 380
      PL2=L2-1.                                                          SIP 390
      W=0.                                                               SIP 400
```

Program listing—Continued

```
      PI=0.                                                             SIP 410
C                                                                       SIP 420
C     ---COMPUTE AVERAGE MAXIMUM PARAMETER FOR PROBLEM---               SIP 430
      DO 10 I=2,INO1                                                    SIP 440
      DO 10 J=2,JNO1                                                    SIP 450
      N=I+DIML*(J-1)                                                    SIP 460
      IF (T(N).EQ.0.) GO TO 10                                          SIP 470
      PI=PI+1.                                                          SIP 480
      DX=DELX(J)/WIDTH                                                  SIP 490
      DY=DELY(I)/YDIM                                                   SIP 500
      W=W+1.-AMIN1(2.*DX*DX/(1.+FACTY*DX*DX/(FACTX*DY*DY)),2.*DY*DY/(1.+SIP 510
     1FACTX*DY*DY/(FACTY*DX*DX)))                                       SIP 520
   10 CONTINUE                                                          SIP 530
      W=W/PI                                                            SIP 540
C                                                                       SIP 550
C     ---COMPUTE PARAMETERS IN GEOMETRIC SEQUENCE---                    SIP 560
      PJ=-1.                                                            SIP 570
      DO 20 I=1,L2                                                      SIP 580
      PJ=PJ+1.                                                          SIP 590
   20 TEMP(I)=1.-(1.-W)**(PJ/PL2)                                       SIP 600
C                                                                       SIP 610
C     ---ORDER SEQUENCE OF PARAMETERS---                                SIP 620
      DO 30 J=1,LENGTH                                                  SIP 630
   30 RHOP(J)=TEMP(IORDER(J))                                           SIP 640
      WRITE (P,370) HMAX                                                SIP 650
      WRITE (P,380) LENGTH,(RHOP(J),J=1,LENGTH)                         SIP 660
      RETURN                                                            SIP 670
C     ..................................................................SIP 680
C                                                                       SIP 690
C     ---INITIALIZE DATA FOR A NEW ITERATION---                         SIP 700
   40 KOUNT=KOUNT+1                                                     SIP 710
      IF (KOUNT.LE.ITMAX) GO TO 50                                      SIP 720
      WRITE (P,360)                                                     SIP 730
      CALL TERM1                                                        SIP 740
   50 IF (MOD(KOUNT,LENGTH)) 60,60,70                                   SIP 750
C     *************************                                         SIP 760
      ENTRY NEWITA                                                      SIP 770
C     *************************                                         SIP 780
   60 NTH=0                                                             SIP 790
   70 NTH=NTH+1                                                         SIP 800
      W=RHOP(NTH)                                                       SIP 810
      TEST3(KOUNT+1)=0.                                                 SIP 820
      TEST=0.                                                           SIP 830
      N=DIML*DIMW                                                       SIP 840
      DO 80 I=1,N                                                       SIP 850
      PHE(I)=PHI(I)                                                     SIP 860
      DEL(I)=0.                                                         SIP 870
      ETA(I)=0.                                                         SIP 880
      V(I)=0.                                                           SIP 890
   80 XI(I)=0.                                                          SIP 900
      BIGI=0.0                                                          SIP 910
C                                                                       SIP 920
C     ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS IN WATER TABLE       SIP 930
C     OR WATER TABLE-ARTESIAN SIMUATION---                              SIP 940
      IF (WATER.NE.CHK(2)) GO TO 90                                     SIP 950
      CALL TRANS                                                        SIP 960
C                                                                       SIP 970
C     ---CHOOSE SIP NORMAL OR REVERSE ALGORITHM---                      SIP 980
   90 IF (MOD(KOUNT,2)) 100,230,100                                     SIP 990
C     ..................................................................SIP1000
C     ---ORDER EQUATIONS WITH ROW 1 FIRST - 3X3 EXAMPLE:                SIP1010
```

Program listing—Continued

```
C              1 2 3                                                    SIP1020
C              4 5 6                                                    SIP1030
C              7 8 9                                                    SIP1040
C     ..................................................................SIP1050
  100 DO 210 I=2,INO1                                                   SIP1060
      DO 210 J=2,JNO1                                                   SIP1070
      N=I+DIML*(J-1)                                                    SIP1080
      NL=N-DIML                                                         SIP1090
      NR=N+DIML                                                         SIP1100
      NA=N-1                                                            SIP1110
      NB=N+1                                                            SIP1120
C                                                                       SIP1130
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---       SIP1140
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 210                           SIP1150
C                                                                       SIP1160
C     ---COMPUTE COEFFICIENTS---                                        SIP1170
      D=TR(NL)/DELX(J)                                                  SIP1180
      F=TR(N)/DELX(J)                                                   SIP1190
      B=TC(NA)/DELY(I)                                                  SIP1200
      H=TC(N)/DELY(I)                                                   SIP1210
      IF (EVAP.NE.CHK(6)) GO TO 120                                     SIP1220
C                                                                       SIP1230
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---              SIP1240
      ETQB=0.                                                           SIP1250
      ETQD=0.0                                                          SIP1260
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 120                           SIP1270
      IF (PHE(N).GT.GRND(N)) GO TO 110                                  SIP1280
      ETQB=QET/ETDIST                                                   SIP1290
      ETQD=ETQB*(ETDIST-GRND(N))                                        SIP1300
      GO TO 120                                                         SIP1310
  110 ETQD=QET                                                          SIP1320
C                                                                       SIP1330
C     ---COMPUTE STORAGE TERM---                                        SIP1340
  120 IF (CONVRT.EQ.CHK(7)) GO TO 130                                   SIP1350
      RHO=S(N)/DELT                                                     SIP1360
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT                               SIP1370
      GO TO 200                                                         SIP1380
C                                                                       SIP1390
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---          SIP1400
  130 SUBS=0.0                                                          SIP1410
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 170             SIP1420
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 160             SIP1430
      IF (KEEP(N)-PHE(N)) 140,150,150                                   SIP1440
  140 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))                           SIP1450
      GO TO 170                                                         SIP1460
  150 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))                           SIP1470
  160 RHO=SY(N)/DELT                                                    SIP1480
      GO TO 180                                                         SIP1490
  170 RHO=S(N)/DELT                                                     SIP1500
  180 IF (LEAK.NE.CHK(9)) GO TO 200                                     SIP1510
C                                                                       SIP1520
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---          SIP1530
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 200                        SIP1540
      HED1=AMAX1(STRT(N),TOP(N))                                        SIP1550
      U=1.                                                              SIP1560
      HED2=0.                                                           SIP1570
      IF (PHE(N).GE.TOP(N)) GO TO 190                                   SIP1580
      HED2=TOP(N)                                                       SIP1590
      U=0.                                                              SIP1600
  190 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))      SIP1610
  200 CONTINUE                                                          SIP1620
```

Program listing—Continued

```
C
C     ---SIP 'NORMAL' ALGORITHM---
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
      E=-B-D-F-H-RHO-TL(N)*U-ETQB
      CH=DEL(NA)*B/(1.+W*DEL(NA))
      GH=ETA(NL)*D/(1.+W*ETA(NL))
      BH=B-W*CH
      DH=D-W*GH
      EH=E+W*CH+W*GH
      FH=F-W*CH
      HH=H-W*GH
      ALFA=BH
      BETA=DH
      GAMA=EH-ALFA*ETA(NA)-BETA*DEL(NL)
      DEL(N)=FH/GAMA
      ETA(N)=HH/GAMA
      RES=-D*PHI(NL)-F*PHI(NR)-H*PHI(NB)-B*PHI(NA)-E*PHI(N)-RHO*KEEP(N)-
     1SL(N)-QRE(N)-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)
      V(N)=(HMAX*RES-ALFA*V(NA)-BETA*V(NL))/GAMA
  210 CONTINUE
C
C     ---BACK SUBSTITUTE FOR VECTOR XI---
      DO 220 I=1,I2
      I3=DIML-I
      DO 220 J=1,J2
      J3=DIMW-J
      N=I3+DIML*(J3-1)
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 220
      XI(N)=V(N)-DEL(N)*XI(N+DIML)-ETA(N)*XI(N+1)
C
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION--
      TCHK=ABS(XI(N))
      IF (TCHK.GT.BIGI) BIGI=TCHK
      PHI(N)=PHI(N)+XI(N)
  220 CONTINUE
      IF (BIGI.GT.ERR) TEST=1.
      TEST3(KOUNT+1)=BIGI
      IF (TEST.EQ.1.) GO TO 40
      RETURN
C
C     ..............................................................
C     ---ORDER EQUATIONS WITH THE LAST ROW FIRST - 3X3  EXAMPLE:
C               7 8 9
C               4 5 6
C               1 2 3
C     ..............................................................
  230 DO 340 II=1,I2
      I=DIML-II
      DO 340 J=2,JNO1
      N=I+DIML*(J-1)
      NL=N-DIML
      NR=N+DIML
      NA=N-1
      NB=N+1
C
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 340
C
C     ---COMPUTE COEFFICIENTS---
      D=TR(NL)/DELX(J)
      F=TR(N)/DELX(J)
```

Program listing—Continued

```
      B=TC(NA)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 250
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 250
      IF (PHE(N).GT.GRND(N)) GO TO 240
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 250
  240 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
  250 IF (CONVRT.EQ.CHK(7)) GO TO 260
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 330
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
  260 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 300
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 290
      IF (KEEP(N)-PHE(N)) 270,280,280
  270 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 300
  280 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  290 RHO=SY(N)/DELT
      GO TO 310
  300 RHO=S(N)/DELT
  310 IF (LEAK.NE.CHK(9)) GO TO 330
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 330
      HED1=AMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 320
      HED2=TOP(N)
      U=0.
  320 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  330 CONTINUE
C
C     ---SIP 'REVERSE' ALGORITHM---
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
      E=-B-D-F-H-RHO-TL(N)*U-ETQB
      CH=DEL(NB)*H/(1.+W*DEL(NB))
      GH=ETA(NL)*D/(1.+W*ETA(NL))
      BH=H-W*CH
      DH=D-W*GH
      EH=E+W*CH+W*GH
      FH=F-W*CH
      HH=B-W*GH
      ALFA=BH
      BETA=DH
      GAMA=EH-ALFA*ETA(NB)-BETA*DEL(NL)
      DEL(N)=FH/GAMA
      ETA(N)=HH/GAMA
      RES=-D*PHI(NL)-F*PHI(NR)-H*PHI(NB)-B*PHI(NA)-E*PHI(N)-RHO*KEEP(N)-
     1SL(N)-QRE(N)-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)
```

```
      V(N)=(HMAX*RES-ALFA*V(NB)-BETA*V(NL))/GAMA
  340 CONTINUE
C
C     ---BACK SUBSTITUTE FOR VECTOR XI---
      DO 350 I3=2,INO1
      DO 350 J=1,J2
      J3=DIMW-J
      N=I3+DIML*(J3-1)
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 350
      XI(N)=V(N)-DEL(N)*XI(N+DIML)-ETA(N)*XI(N-1)
C
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION--
      TCHK=ABS(XI(N))
      IF (TCHK.GT.BIGI) BIGI=TCHK
      PHI(N)=PHI(N)+XI(N)
  350 CONTINUE
      IF (BIGI.GT.ERR) TEST=1.
      TEST3(KOUNT+1)=BIGI
      IF (TEST.EQ.1.) GO TO 40
      RETURN
C
C     ..................................................................
C
C     ---FORMATS---
C
C     ------------------------------------------------------------------
C
C
  360 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))
  370 FORMAT ('-',44X,'SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE'/45X,
     143('_'),//,61X,'BETA=',F5.2)
  380 FORMAT (1H0,I5,22H ITERATION PARAMETERS:,6D15.7/(/28X,6D15.7/))
      END
```

```
      SUBROUTINE SOLVE2(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,D
     1EL,ETA,V,XI,DELX,BETA,DELY,ALFA,TEST3,TR,TC,GRND,SY,TOP,RATE,M,RIV
     2ER)
C     ------------------------------------------------------------------
C     SOLUTION BY LINE SUCCESSIVE OVERRELAXATION
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI,DBLE,RHOP(20),G,BE,TEMP,IMK,DABS,W,PARAM,TEST2,DMAX1,B2
     1,A,C,B1,E,Q,RHO,B,D,F,H
      REAL *4KEEP,M
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
C
      DIMENSION PHI(1), BE(1), G(1), TEMP(1), KEEP(1), PHE(1), STRT(1),
     1T(1), S(1), QRE(1), WELL(1), TL(1), SL(1), DEL(1), ETA(1), V(1), X
     2I(1), DELX(1), BETA(1), DELY(1), ALFA(1), TEST3(1), TR(1), TC(1),
     3GRND(1), SY(1), TOP(1), RATE(1), M(1), RIVER(1)
C
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LE
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DI
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2
      RETURN
```

Program listing—Continued

```
C     ..................................................................SOR 270
C                                                                       SOR 280
C     ---WRITE ACCELERATION PARAMETER---                                SOR 290
C     **********************                                            SOR 300
      ENTRY ITER2                                                       SOR 310
C     **********************                                            SOR 320
      WRITE (P,490)                                                     SOR 330
      WRITE (P,500) HMAX,LENGTH                                         SOR 340
      RETURN                                                            SOR 350
C     ..................................................................SOR 360
C                                                                       SOR 370
C     ---INITIALIZE DATA FOR A NEW ITERATION---                         SOR 380
   10 KOUNT=KOUNT+1                                                     SOR 390
      IF (KOUNT.LE.ITMAX) GO TO 20                                      SOR 400
      WRITE (P,510)                                                     SOR 410
      CALL TERM1                                                        SOR 420
C     ************************                                          SOR 430
      ENTRY NEWITB                                                      SOR 440
C     ************************                                          SOR 450
   20 TEST3(KOUNT+1)=0.                                                 SOR 460
      TEST=0.                                                           SOR 470
      N=DIML*DIMW                                                       SOR 480
      DO 30 I=1,N                                                       SOR 490
   30 PHE(I)=PHI(I)                                                     SOR 500
      BIGI=0.0                                                          SOR 510
C                                                                       SOR 520
C     ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS IN WATER TABLE       SOR 530
C     OR WATER TABLE-ARTESIAN SIMUATION---                              SOR 540
      IF (WATER.NE.CHK(2)) GO TO 40                                     SOR 550
      CALL TRANS                                                        SOR 560
C     ..................................................................SOR 570
C                                                                       SOR 580
C     ---SOLUTION BY LSOR---                                            SOR 590
C     ----------------------                                            SOR 600
   40 NO3=DIMW-2                                                        SOR 610
      TEMP(DIMW)=0.0                                                    SOR 620
      DO 170 I=2,INO1                                                   SOR 630
      DO 150 J=2,JNO1                                                   SOR 640
      N=I+DIML*(J-1)                                                    SOR 650
      NA=N-1                                                            SOR 660
      NB=N+1                                                            SOR 670
      NL=N-DIML                                                         SOR 680
      NR=N+DIML                                                         SOR 690
      BE(J)=0.0                                                         SOR 700
      G(J)=0.0                                                          SOR 710
C                                                                       SOR 720
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---       SOR 730
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 150                           SOR 740
C                                                                       SOR 750
C     ---COMPUTE COEFFICIENTS---                                        SOR 760
      D=TR(N-DIML)/DELX(J)                                              SOR 770
      F=TR(N)/DELX(J)                                                   SOR 780
      B=TC(N-1)/DELY(I)                                                 SOR 790
      H=TC(N)/DELY(I)                                                   SOR 800
      IF (EVAP.NE.CHK(6)) GO TO 60                                      SOR 810
C                                                                       SOR 820
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---              SOR 830
      ETQB=0.                                                           SOR 840
      ETQD=0.0                                                          SOR 850
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 60                            SOR 860
      IF (PHE(N).GT.GRND(N)) GO TO 50                                   SOR 870
```

Program listing—Continued

```
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 60
   50 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
   60 IF (CONVRT.EQ.CHK(7)) GO TO 70
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 140
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
   70 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 110
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 100
      IF (KEEP(N)-PHE(N)) 80,90,90
   80 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 110
   90 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  100 RHO=SY(N)/DELT
      GO TO 120
  110 RHO=S(N)/DELT
  120 IF (LEAK.NE.CHK(9)) GO TO 140
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 140
      HED1=AMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 130
      HED2=TOP(N)
      U=0.
  130 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  140 CONTINUE
C
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR G---
      E=-D-F-B-H-RHO-TL(N)*U-ETQB
      W=E-D*BE(J-1)
      BE(J)=F/W
      Q=-B*PHI(NA)-H*PHI(NB)-RHO*KEEP(N)-SL(N)-QRE(N)-WELL(N)+ETQD-SUBS-
     1TL(N)*STRT(N)-D*PHI(NL)-F*PHI(NR)-E*PHI(N)
      G(J)=(Q-D*G(J-1))/W
  150 CONTINUE
C
C     ---BACK SUBSTITUTE FOR TEMP---
      DO 160 KNO4=1,NO3
      NO4=DIMW-KNO4
      TEMP(NO4)=G(NO4)-BE(NO4)*TEMP(NO4+1)
  160 CONTINUE
C
C     ---EXTRAPOLATED VALUE OF PHI---
      DO 170 J=2,JNO1
      N=I+DIML*(J-1)
      PHI(N)=PHI(N)+HMAX*TEMP(J)
C
C     ---COMPARE DIFFERENCE WITH CLOSURE CRITERION--
      TCHK=DABS(TEMP(J))
      IF (TCHK.GT.BIGI) BIGI=TCHK
  170 CONTINUE
      IF (BIGI.GT.ERR) TEST=1.
      TEST3(KOUNT+1)=BIGI
```

Program listing—Continued

```
      IF (KOUNT.EQ.0) GO TO 10
      IF (TEST.EQ.0.) RETURN
C
C     ---TEST FOR TWO DIMENSIONAL CORRECTION---
      IF (MOD(KOUNT,LENGTH).NE.0) GO TO 10
      GO TO 200
  180 DO 190 I=2,INO1
      DO 190 J=2,JNO1
      N=I+DIML*(J-1)
      IF (T(N).EQ.0.) GO TO 190
      PHI(N)=PHI(N)+ALFA(I)+BETA(J)
  190 CONTINUE
      GO TO 10
C     ..................................................................
C
C     ---TWO DIMENSIONAL CORRECTION TO LSOR---
C     ----------------------------------------
C
C     ---COMPUTE ALFA CORRECTION FOR ROWS---
  200 DO 210 I=1,DIML
      ALFA(I)=0.
      BE(I)=0.0
  210 G(I)=0.0
      DO 330 I=2,INO1
      A=0.
      B2=0.
      C=0.
      Q=0.
C
C     ---SUMMATION OF COEFFICIENTS FOR EACH ROW---
      DO 320 J=2,JNO1
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
      NR=N+DIML
      IF (S(N).LT.0.) GO TO 330
      IF (T(N).EQ.0.) GO TO 320
C
C     ---COMPUTE COEFFICIENTS---
      D=TR(N-DIML)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 230
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 230
      IF (PHE(N).GT.GRND(N)) GO TO 220
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 230
  220 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
  230 IF (CONVRT.EQ.CHK(7)) GO TO 240
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 310
```

Program listing—Continued

```
  240 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 280
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 270
      IF (KEEP(N)-PHE(N)) 250,260,260
  250 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 280
  260 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  270 RHO=SY(N)/DELT
      GO TO 290
  280 RHO=S(N)/DELT
  290 IF (LEAK.NE.CHK(9)) GO TO 310
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 310
      HED1=AMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 300
      HED2=TOP(N)
      U=0.
  300 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  310 CONTINUE
C
      A=A-B
      B1=B+H+RHO+TL(N)*U+ETQB
      B2=B2+B1
      C=C-H
      Q=Q+(D*PHI(NL)+F*PHI(NR)+B*PHI(NA)+H*PHI(NB)+RHO*KEEP(N)+SL(N)+QRE
     1(N)+WELL(N)-ETQD+SUBS+TL(N)*STRT(N)-(D+F+B1)*PHI(N))
  320 CONTINUE
C
C     ---COMPUTATION OF INTERMEDIATE VECTOR G---
      W=B2-A*BE(I-1)
      BE(I)=C/W
      G(I)=(Q-A*G(I-1))/W
  330 CONTINUE
C
C     ---BACK SUBSTITUTE FOR ALFA---
      NO3=DIML-2
      DO 340 KNO4=1,NO3
      NO4=DIML-KNO4
  340 ALFA(NO4)=G(NO4)-BE(NO4)*ALFA(NO4+1)
C     ******************************************************************
C
C     ---COMPUTE BETA CORRECTION FOR COLUMNS---
      DO 350 J=1,DIMW
      BETA(J)=0.
      BE(J)=0.0
  350 G(J)=0.0
      DO 470 J=2,JNO1
      A=0.
      B2=0.
      C=0.
      Q=0.
C
C     ---SUMMATION OF COEFFICIENTS FOR EACH COLUMN---
      DO 460 I=2,INO1
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
```

Program listing—Continued

```
      NR=N+DIML
      IF (S(N).LT.0.) GO TO 470
      IF (T(N).EQ.0.) GO TO 460
      D=TR(N-DIML)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 370
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 370
      IF (PHE(N).GT.GRND(N)) GO TO 360
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 370
  360 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
  370 IF (CONVRT.EQ.CHK(7)) GO TO 380
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 450
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
  380 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 420
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 410
      IF (KEEP(N)-PHE(N)) 390,400,400
  390 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 420
  400 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  410 RHO=SY(N)/DELT
      GO TO 430
  420 RHO=S(N)/DELT
  430 IF (LEAK.NE.CHK(9)) GO TO 450
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 450
      HED1=AMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 440
      HED2=TOP(N)
      U=0.
  440 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  450 CONTINUE
C
      A=A-D
      B1=D+F+RHO+TL(N)*U+ETQB
      B2=B2+B1
      C=C-F
      Q=Q+(D*PHI(NL)+F*PHI(NR)+B*PHI(NA)+H*PHI(NB)+RHO*KEEP(N)+SL(N)+QRE
     1(N)+WELL(N)-ETQD+SUBS+TL(N)*STRT(N)-(B+H+B1)*PHI(N))
  460 CONTINUE
C
C     ---COMPUTATION OF INTERMEDIATE VECTOR G---
      W=B2-A*BE(J-1)
      BE(J)=C/W
      G(J)=(Q-A*G(J-1))/W
```

Program listing—Continued

```
 470 CONTINUE
C
C     ---BACK SUBSTITUTE FOR BETA---
      NO3=DIMW-2
      DO 480 KNO4=1,NO3
      NO4=DIMW-KNO4
 480 BETA(NO4)=G(NO4)-BE(NO4)*BETA(NO4+1)
      GO TO 180
C     ..................................................................
C
C     ---FORMATS---
C
C     ------------------------------------------------------------------
C
C
 490 FORMAT ('-',45X,'SOLUTION BY LINE SUCCESSIVE OVERRELAXATION'/46X,4
     12('_'))
 500 FORMAT ('-',26X,'ACCELERATION PARAMETER =',F6.3,'  TWO DIMENSIONAL
     1 CORRECTION EVERY',I5,' ITERATIONS')
 510 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))
      END
      SUBROUTINE SOLVE3(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,D
     1EL,ETA,V,XI,DELX,BETA,DELY,ALFA,XII,TEST3,TR,TC,GRND,SY,TOP,RATE,M
     2,RIVER)
C     ------------------------------------------------------------------
C     SOLUTION BY THE ALTERNATING DIRECTION IMPLICIT PROCEDURE
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      REAL *8PHI,DBLE,RHOP(20),G,BE,TEMP,IMK,DABS,W,PARAM,TEST2,DMAX1,DT
     1ERMS,B1,E,Q,B,D,F,H,RHO,XII
      REAL *4KEEP,M
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
C
      DIMENSION PHI(1), BE(1), G(1), TEMP(1), KEEP(1), PHE(1), STRT(1),
     1T(1), S(1), QRE(1), WELL(1), TL(1), SL(1), DEL(1), ETA(1), V(1), X
     2I(1), DELX(1), BETA(1), DELY(1), ALFA(1), XII(1), TEST3(1), TR(1),
     3 TC(1), GRND(1), SY(1), TOP(1), RATE(1), M(1), RIVER(1)
C
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LE
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DI
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2
      RETURN
C     ..................................................................
C
C     ---COMPUTE AND PRINT ITERATION PARAMETERS---
C     ********************
      ENTRY ITER3
C     ********************
      HMIN=2.
      IN4=DIMW-2
      IN5=DIML-2
      XVAL=3.1415**2/(2.*IN4*IN4)
      YVAL=3.1415**2/(2.*IN5*IN5)
      DO 10 I=2,INO1
```

```
      DO 10 J=2,JNO1
      N=I+DIML*(J-1)
      IF (T(N).EQ.0.) GO TO 10
      XPART=XVAL*(1/(1+DELX(J)**2*FACTY/DELY(I)**2*FACTX))
      YPART=YVAL*(1/(1+DELY(I)**2*FACTX/DELX(J)**2*FACTY))
      HMIN=AMIN1(HMIN,XPART,YPART)
   10 CONTINUE
      ALPHA=EXP(ALOG(HMAX/HMIN)/(LENGTH-1))
      RHOP(1)=HMIN
      DO 20 NTIME=2,LENGTH
   20 RHOP(NTIME)=RHOP(NTIME-1)*ALPHA
      WRITE (P,400)
      WRITE (P,410) LENGTH,(RHOP(J),J=1,LENGTH)
      RETURN
C     ..................................................................
C
C     ---INITIALIZE DATA FOR A NEW ITERATION---
   30 KOUNT=KOUNT+1
      IF (KOUNT.LE.ITMAX) GO TO 40
      WRITE (P,390)
      CALL TERM1
   40 IF (MOD(KOUNT,LENGTH)) 50,50,60
C     ************************
      ENTRY NEWITC
C     ************************
   50 NTH=0
   60 NTH=NTH+1
      PARAM=RHOP(NTH)
      TEST3(KOUNT+1)=0.
      TEST=0.
      N=DIML*DIMW
      DO 70 I=1,N
   70 PHE(I)=PHI(I)
      BIGI=0.0
C
C     ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS IN WATER TABLE
C     OR WATER TABLE-ARTESIAN SIMUATION---
      IF (WATER.NE.CHK(2)) GO TO 80
      CALL TRANS
C     ..................................................................
C
C     ---SOLUTION BY ADI---
C     ---------------------
C     ---COMPUTE IMPLICITLY ALONG ROWS---
   80 NO3=DIMW-2
      DO 90 J=1,DIMW
      N=1+DIML*(J-1)
   90 TEMP(J)=PHI(N)
      DO 230 I=2,DIML
      DO 200 J=2,JNO1
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
      NR=N+DIML
      BE(J)=0.0
      G(J)=0.0
C
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 200
C
```

Program listing—Continued

```
C     ---COMPUTE COEFFICIENTS---
      D=TR(N-DIML)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 110
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 110
      IF (PHE(N).GT.GRND(N)) GO TO 100
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 110
  100 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
  110 IF (CONVRT.EQ.CHK(7)) GO TO 120
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 190
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
  120 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 160
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 150
      IF (KEEP(N)-PHE(N)) 130,140,140
  130 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 160
  140 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  150 RHO=SY(N)/DELT
      GO TO 170
  160 RHO=S(N)/DELT
  170 IF (LEAK.NE.CHK(9)) GO TO 190
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 190
      HED1=AMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 180
      HED2=TOP(N)
      U=0.
  180 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  190 CONTINUE
C
C     ---CALCULATE VALUES FOR PARAMETERS USED IN THOMAS ALGORITHM
C     AND FORWARD SUBSTITUTE TO COMPUTE INTERMEDIATE VECTOR G---
      IMK=(B+D+F+H)*PARAM
      E=-D-F-RHO-IMK-TL(N)*U-ETQB
      W=E-D*BE(J-1)
      BE(J)=F/W
      Q=-B*PHI(NA)+(B+H-IMK-E)*PHI(N)-H*PHI(NB)-RHO*KEEP(N)-SL(N)-QRE(N)
     1-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)-D*PHI(NL)-F*PHI(NR)
      G(J)=(Q-D*G(J-1))/W
  200 CONTINUE
C
C     ---BACK SUBSTITUTE FOR HEAD VALUES AND PLACE THEM IN TEMP---
      XII(DIMW)=0.D0
      DO 220 KN04=1,N03
```

Program listing—Continued

```
      NO4=DIMW-KNO4                                                     ADI1610
      N=I+DIML*(NO4-1)                                                  ADI1620
C                                                                       ADI1630
C     ---FIRST PLACE TEMP VALUES IN PHI(N-1)---                         ADI1640
      PHI(N-1)=TEMP(NO4)                                                ADI1650
      IF (T(N).NE.0..AND.S(N).GE.0.) GO TO 210                          ADI1660
      XII(NO4)=0.D0                                                     ADI1670
      GO TO 220                                                         ADI1680
  210 XII(NO4)=G(NO4)-BE(NO4)*XII(NO4+1)                                ADI1690
  220 TEMP(NO4)=PHI(N)+XII(NO4)                                         ADI1700
  230 CONTINUE                                                          ADI1710
C     ..................................................................ADI1720
C                                                                       ADI1730
C     ---COMPUTE IMPLICITLY ALONG COLUMNS---                            ADI1740
      NO3=DIML-2                                                        ADI1750
      DO 240 I=1,DIML                                                   ADI1760
  240 TEMP(I)=PHI(I)                                                    ADI1770
      DO 380 J=2,DIMW                                                   ADI1780
      DO 350 I=2,INO1                                                   ADI1790
      N=I+DIML*(J-1)                                                    ADI1800
      NA=N-1                                                            ADI1810
      NB=N+1                                                            ADI1820
      NL=N-DIML                                                         ADI1830
      NR=N+DIML                                                         ADI1840
      BE(I)=0.0                                                         ADI1850
      G(I)=0.0                                                          ADI1860
C                                                                       ADI1870
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---       ADI1880
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 350                           ADI1890
C                                                                       ADI1900
C     ---COMPUTE COEFFICIENTS---                                        ADI1910
      D=TR(N-DIML)/DELX(J)                                              ADI1920
      F=TR(N)/DELX(J)                                                   ADI1930
      B=TC(N-1)/DELY(I)                                                 ADI1940
      H=TC(N)/DELY(I)                                                   ADI1950
      IF (EVAP.NE.CHK(6)) GO TO 260                                     ADI1960
C                                                                       ADI1970
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---              ADI1980
      ETQB=0.                                                           ADI1990
      ETQD=0.0                                                          ADI2000
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 260                           ADI2010
      IF (PHE(N).GT.GRND(N)) GO TO 250                                  ADI2020
      ETQB=QET/ETDIST                                                   ADI2030
      ETQD=ETQB*(ETDIST-GRND(N))                                        ADI2040
      GO TO 260                                                         ADI2050
  250 ETQD=QET                                                          ADI2060
C                                                                       ADI2070
C     ---COMPUTE STORAGE TERM---                                        ADI2080
  260 IF (CONVRT.EQ.CHK(7)) GO TO 270                                   ADI2090
      RHO=S(N)/DELT                                                     ADI2100
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT                               ADI2110
      GO TO 340                                                         ADI2120
C                                                                       ADI2130
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---          ADI2140
  270 SUBS=0.0                                                          ADI2150
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 310             ADI2160
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 300             ADI2170
      IF (KEEP(N)-PHE(N)) 280,290,290                                   ADI2180
  280 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))                           ADI2190
      GO TO 310                                                         ADI2200
  290 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))                           ADI2210
```

Program listing—Continued

```
  300 RHO=SY(N)/DELT
      GO TO 320
  310 RHO=S(N)/DELT
  320 IF (LEAK.NE.CHK(9)) GO TO 340
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 340
      HED1=AMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 330
      HED2=TOP(N)
      U=0.
  330 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  340 CONTINUE
C
C     ---CALCULATE VALUES FOR PARAMETERS USED IN THOMAS ALGORITHM
C     AND FORWARD SUBSTITUTE TO COMPUTE INTERMEDIATE VECTOR G---
      IMK=(B+D+F+H)*PARAM
      E=-B-H-RHO-IMK-TL(N)*U-ETQB
      W=E-B*BE(I-1)
      BE(I)=H/W
      Q=-D*PHI(NL)+(D+F-IMK-E)*PHI(N)-F*PHI(NR)-RHO*KEEP(N)-SL(N)-QRE(N)
     1-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)-B*PHI(NA)-H*PHI(NB)
      G(I)=(Q-B*G(I-1))/W
  350 CONTINUE
C
C     ---BACK SUBSTITUTE FOR HEAD VALUES AND PLACE THEM IN TEMP---
      XII(DIML)=0.D0
      DO 370 KNO4=1,NO3
      NO4=DIML-KNO4
      N=NO4+DIML*(J-1)
C
C     ---FIRST PLACE TEMP VALUES IN PHI(N-DIML)----
      PHI(N-DIML)=TEMP(NO4)
      IF (T(N).NE.0..AND.S(N).GE.0.) GO TO 360
      XII(NO4)=0.D0
      TEMP(NO4)=PHI(N)
      GO TO 370
  360 XII(NO4)=G(NO4)-BE(NO4)*XII(NO4+1)
      TEMP(NO4)=PHI(N)+XII(NO4)
C
C     ---COMPARE CHANGE IN HEAD WITH CLOSURE CRITERION--
      TCHK=ABS(SNGL(TEMP(NO4))-PHE(N))
      IF (TCHK.GT.BIGI) BIGI=TCHK
  370 CONTINUE
  380 CONTINUE
      IF (BIGI.GT.ERR) TEST=1.
      TEST3(KOUNT+1)=BIGI
      IF (TEST.EQ.1.) GO TO 30
      RETURN
C     ..................................................................
C
C     ---FORMATS---
C
C     ------------------------------------------------------------------
C
C
  390 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))
  400 FORMAT ('-',38X,'SOLUTION BY THE ALTERNATING DIRECTION IMPLICIT PR
     1OCEDURE'/39X,56('_'))
```

Program listing—Continued

```
  410 FORMAT (///1H0,I5,22H ITERATION PARAMETERS:,8D12.3//28X,10D12.3)  ADI2830
      END                                                               ADI2840-

      SUBROUTINE COEF(PHI,KEEP,PHE,STRT,SURI,T,TR,TC,S,WELL,TL,SL,PERM,BCOF   10
     1OTTOM,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY)                         COF   20
C     ------------------------------------------------------------------COF   30
C     COMPUTE COEFFICIENTS                                              COF   40
C     ------------------------------------------------------------------COF   50
C                                                                       COF   60
C     SPECIFICATIONS:                                                   COF   70
      REAL *8PHI,DBLE,RHO,B,D,F,H                                       COF   80
      REAL *4KEEP,M                                                     COF   90
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,COF  100
     1CCNTR,LEAK,RECH,SIP,ADI                                           COF  110
C                                                                       COF  120
      DIMENSION PHI(1), KEEP(1), PHE(1), STRT(1), SURI(1), T(1), TR(1), COF  130
     1TC(1), S(1), WELL(1), TL(1), SL(1), PERM(1), BOTTOM(1), SY(1), RATCOF  140
     2E(1), RIVER(1), M(1), TOP(1), GRND(1), DELX(1), DELY(1)           COF  150
C                                                                       COF  160
      COMMON /SARRAY/ VF4(11),CHK(15)                                   COF  170
      COMMON /SPARAM/ WATER,CCNVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LECOF  180
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,COF  190
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,COF  200
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DICOF  210
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2                                 COF  220
C                                                                       COF  230
      DATA PIE/3.141593/                                                COF  235
      RETURN                                                            COF  240
C     ..................................................................COF  250
C                                                                       COF  260
C     ---COMPUTE COEFFICIENTS FOR TRANSIENT PART OF LEAKAGE TERM---      COF  270
C     ********************                                              COF  280
      ENTRY CLAY                                                        COF  290
C     ********************                                              COF  300
      TMIN=1.E30                                                        COF  310
      TT=0.0                                                            COF  320
      PRATE=0.                                                          COF  330
      DC 50 I=1,DIML                                                    COF  340
      DC 50 J=1,DIMW                                                    COF  350
      N=I+DIML*(J-1)                                                    COF  360
C                                                                       COF  370
C     ---SKIP COMPUTATICNS IF T, RATE OR M = 0, OR IF CONSTANT          COF  380
C        HEAD BOUNDARY---                                               COF  390
      IF (RATE(N).LE.0..OR.T(N).EQ.0..OR.M(N).EQ.0..OR.S(N).LT.0.) GO TOCOF  400
     1 50                                                               COF  410
C                                                                       COF  420
C     ---IF VALUE FOR TL(N  ) WILL EQUAL VALUE FOR PREVIOUS NODE,       COF  430
C     SKIP PART OF COMPUTATIONS---                                      COF  440
      IF (RATE(N)*M(N).EQ.PRATE) GO TO 40                               COF  450
      DIMT=RATE(N)*SUMP/(M(N)*M(N)*SS*3)                                COF  460
      IF (DIMT.GT.TT) TT=DIMT                                           COF  470
      IF (DIMT.LT.TMIN) TMIN=DIMT                                       COF  480
      PPT=PIE*PIE*DIMT                                                  COF  490
C                                                                       COF  500
C     ---RECOMPUTE PPT IF DIMT WITHIN RANGE FOR SHORT TIME COMPUTATION--COF  510
      IF (DIMT.LT.1.0E-03) PPT=1.0/DIMT                                 COF  520
      CC=(2.3-PPT)/(2.*PPT)                                             COF  530
C                                                                       COF  540
C     ---COMPUTE SUM OF EXPONENTIALS---                                 COF  550
      SUMN=0.0                                                          COF  560
```

Program listing—Continued

```
      DO 20 K=1,200
      POWER=K*K*PPT
      IF (POWER.LE.150.) GO TO 10
      POWER=150
   10 PEX=EXP(-POWER)
      SUMN=SUMN+PEX
      IF (PEX.GT.0.00009) GO TO 20
      IF (K.GT.CC) GO TO 30
   20 CONTINUE
C
C     ---COMPUTE DENOMINATER DEPENDING ON VALUE OF DIMT---
   30 DENOM=1.0
      IF (DIMT.LT.1.0E-03) DENOM=SQRT(PIE*DIMT)
C
C     ---HEAD VALUES ARE NOT INCLUDED IN COMPUTATION OF Q FACTOR SINCE
C        LEAKAGE IS CONSIDERED IMPLICITLY---
   40 Q1=RATE(N)/(M(N)*DENOM)
      TL(N)=Q1+2.*Q1*SUMN
      PRATE=RATE(N)*M(N)
   50 CONTINUE
      TMIN=TMIN*3.0
      TT=TT*3.0
      RETURN
C     ..................................................................
C
C     ---COMPUTE TRANSMISSIVITY IN WT OR WT-ARTESIAN CONVERSION PROBLEM-
C     ********************
      ENTRY TRANS
C     ********************
      DO 60 I=1,DIML
      DO 60 J=1,DIMW
      N=I+DIML*(J-1)
      IF (PERM(N).EQ.0.) GO TO 60
      HED=PHI(N)
      IF (CONVRT.EQ.CHK(7)) HED=AMIN1(SNGL(PHI(N)),TOP(N))
      T(N)=PERM(N)*(HED-BOTTOM(N))
      IF (T(N).GT.0.) GO TO 60
      IF (WELL(N).LT.0.) GO TO 70
C
C     ---THE FOLLOWING STATEMENTS APPLY WHEN NODES (EXCEPT WELL NODES)
C     GO DRY---
      PERM(N)=0.
      T(N)=0.0
      TR(N-DIML)=0.
      TR(N)=0.
      TC(N-1)=0.
      TC(N)=0.
      PHI(N)=SURI(N)
      WRITE (P,150) I,J
   60 CONTINUE
      IF (KT.EQ.0) RETURN
      GO TO 90
C
C     ---START PROGRAM TERMINATION WHEN A WELL GOES DRY---
   70 WRITE (P,120) I,J
      WRITE (P,130)
      IERR=1
      CALL DRDN
      DO 80 I=2,INO1
      DO 80 J=2,JNO1
      N=I+DIML*(J-1)
```

Program listing—Continued

```
   80 PHI(N)=KEEP(N)                                                    COF1180
      SUM=SUM-DELT                                                      COF1190
      SUMP=SUMP-DELT                                                    COF1200
      KT=KT-1                                                           COF1210
      IF (KT.EQ.0) STOP                                                 COF1220
      IF (IDK2.EQ.CHK(15)) CALL DISK                                    COF1230
      IF (PNCH.EQ.CHK(1)) CALL PUNCH                                    COF1240
      IF (MOD(KT,KTH).EQ.0) STOP                                        COF1250
      WRITE (P,140) KT,SUM                                              COF1260
      CALL DRDN                                                         COF1270
      IF (CHCK.EQ.CHK(5)) CALL CWRITE                                   COF1280
      STOP                                                              COF1290
C                                                                       COF1300
C     ---COMPUTE T COEFFICIENTS---                                      COF1310
C     ********************                                              COF1320
      ENTRY TCOF                                                        COF1330
C     ********************                                              COF1340
   90 DO 110 I=1,INO1                                                   COF1350
      DO 110 J=1,JNO1                                                   COF1360
      N=I+DIML*(J-1)                                                    COF1370
      NR=N+DIML                                                         COF1380
      NB=N+1                                                            COF1390
      IF (T(N).EQ.0.) GO TO 110                                         COF1400
      IF (T(NR).EQ.0.) GO TO 100                                        COF1410
      TR(N)=(2.*T(NR)*T(N))/(T(N)*DELX(J+1)+T(NR)*DELX(J))*FACTX        COF1420
  100 IF (T(NB).EQ.0.) GO TO 110                                        COF1430
      TC(N)=(2.*T(NB)*T(N))/(T(N)*DELY(I+1)+T(NB)*DELY(I))*FACTY        COF1440
  110 CONTINUE                                                          COF1450
      RETURN                                                            COF1460
C                                                                       COF1470
C     ---FORMATS---                                                     COF1480
C                                                                       COF1490
C     ------------------------------------------------------------------COF1500
C                                                                       COF1510
C                                                                       COF1520
  120 FORMAT ('-*************WELL',I3,',',I3,' GOES DRY**************') COF1530
  130 FORMAT ('1',50X,'DRAWDOWN WHEN WELL WENT DRY')                    COF1540
  140 FORMAT ('1',32X,'DRAWDOWN FOR TIME STEP',I3,'; SIMULATION TIME =',COF1550
     11PE15.7,' SECONDS')                                               COF1560
  150 FORMAT ('-',20('*'),' NODE ',I4,',',I4,' GOES DRY ',20('*'))      COF1570
      END                                                               COF1580-


      SUBROUTINE CHECKI(PHI,KEEP,PHE,STRT,T,TR,TC,S,QRE,WELL,TL,PERM,BOTCHK  10
     1TOM,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY)                           CHK  20
C     ------------------------------------------------------------------CHK  30
C     COMPUTE A MASS BALANCE                                            CHK  40
C     ------------------------------------------------------------------CHK  50
C                                                                       CHK  60
C     SPECIFICATIONS:                                                   CHK  70
      REAL *8PHI,DBLE                                                   CHK  80
      REAL *4KEEP,M                                                     CHK  90
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CHK 100
     1CONTR,LEAK,RECH,SIP,ADI                                           CHK 110
C                                                                       CHK 120
      DIMENSION PHI(IZ,JZ), KEEP(IZ,JZ), PHE(IZ,JZ), STRT(IZ,JZ), T(IZ,JCHK 130
     1Z), TR(IZ,JZ), TC(IZ,JZ), S(IZ,JZ), QRE(IZ,JZ), WELL(IZ,JZ), TL(IZCHK 140
     2,JZ), PERM(IZ,JZ), BOTTOM(IP,JP), SY(IP,JP), RATE(IR,JR), RIVER(IRCHK 150
     3,JR), M(IR,JR), TOP(IC,JC), GRND(IL,JL), DELX(JZ), DELY(IZ)       CHK 160
C                                                                       CHK 170
      COMMON /SARRAY/ VF4(11),CHK(15)                                   CHK 180
```

Program listing—Continued

```
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LECHK 190
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,CHK 200
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,CHK 210
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DICHK 220
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2                                 CHK 230
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT  CHK 240
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1  CHK 250
      RETURN                                                            CHK 260
C     ..................................................................CHK 270
C     ********************                                              CHK 280
      ENTRY CHECK                                                       CHK 290
C     ********************                                              CHK 300
C     ---INITIALIZE VARIABLES---                                        CHK 310
      PUMP=0.                                                           CHK 320
      STOR=0.                                                           CHK 330
      FLUXS=0.0                                                         CHK 340
      CHD1=0.0                                                          CHK 350
      CHD2=0.0                                                          CHK 360
      QREFLX=0.                                                         CHK 370
      CFLUX=0.                                                          CHK 380
      FLUX=0.                                                           CHK 390
      ETFLUX=0.                                                         CHK 400
      FLXN=0.0                                                          CHK 410
C     ..................................................................CHK 420
C                                                                       CHK 430
C     ---COMPUTE RATES,STORAGE AND PUMPAGE FOR THIS STEP---             CHK 440
      DO 240 I=2,DIML                                                   CHK 450
      DO 240 J=2,DIMW                                                   CHK 460
      IF (T(I,J).EQ.0.) GO TO 240                                       CHK 470
      AREA=DELX(J)*DELY(I)                                              CHK 480
      IF (S(I,J).GE.0.) GO TO 120                                       CHK 490
C                                                                       CHK 500
C     ---COMPUTE FLOW RATES TO AND FROM CONSTANT HEAD BOUNDARIES---     CHK 510
      IF (S(I,J-1).LT.0..OR.T(I,J-1).EQ.0.) GO TO 30                    CHK 520
      X=(STRT(I,J)-PHI(I,J-1))*TR(I,J-1)*DELY(I)                        CHK 530
      IF (X) 10,30,20                                                   CHK 540
   10 CHD1=CHD1+X                                                       CHK 550
      GO TO 30                                                          CHK 560
   20 CHD2=CHD2+X                                                       CHK 570
   30 IF (S(I,J+1).LT.0..OR.T(I,J+1).EQ.0.) GO TO 60                    CHK 580
      X=(STRT(I,J)-PHI(I,J+1))*TR(I,J)*DELY(I)                          CHK 590
      IF (X) 40,60,50                                                   CHK 600
   40 CHD1=CHD1+X                                                       CHK 610
      GO TO 60                                                          CHK 620
   50 CHD2=CHD2+X                                                       CHK 630
   60 IF (S(I-1,J).LT.0..OR.T(I-1,J).EQ.0.) GO TO 90                    CHK 640
      X=(STRT(I,J)-PHI(I-1,J))*TC(I-1,J)*DELX(J)                        CHK 650
      IF (X) 70,90,80                                                   CHK 660
   70 CHD1=CHD1+X                                                       CHK 670
      GO TO 90                                                          CHK 680
   80 CHD2=CHD2+X                                                       CHK 690
   90 IF (S(I+1,J).LT.0..OR.T(I+1,J).EQ.0.) GO TO 240                   CHK 700
      X=(STRT(I,J)-PHI(I+1,J))*TC(I,J)*DELX(J)                          CHK 710
      IF (X) 100,240,110                                                CHK 720
  100 CHD1=CHD1+X                                                       CHK 730
      GO TO 240                                                         CHK 740
  110 CHD2=CHD2+X                                                       CHK 750
      GO TO 240                                                         CHK 760
C                                                                       CHK 770
C     ---RECHARGE AND WELLS---                                          CHK 780
  120 QREFLX=QREFLX+QRE(I,J)*AREA                                       CHK 790
```

Program listing—Continued

```
      IF (WELL(I,J)) 130,150,140
  130 PUMP=PUMP+WELL(I,J)*AREA
      GO TO 150
  140 CFLUX=CFLUX+WELL(I,J)*AREA
  150 IF (EVAP.NE.CHK(6)) GO TO 190
C
C     ---COMPUTE ET RATE---
      IF (PHI(I,J).GE.GRND(I,J)-ETDIST) GO TO 160
      ETQ=0.0
      GO TO 180
  160 IF (PHI(I,J).LE.GRND(I,J)) GO TO 170
      ETQ=QET
      GO TO 180
  170 ETQ=QET/ETDIST*(PHI(I,J)+ETDIST-GRND(I,J))
  180 ETFLUX=ETFLUX-ETQ*AREA
C
C     ---COMPUTE VOLUME FROM STORAGE---
  190 STORE=S(I,J)
      IF (WATER.EQ.CHK(2)) STORE=SY(I,J)
      IF (CONVRT.NE.CHK(7)) GO TO 230
      X=KEEP(I,J)-PHI(I,J)
      IF (X) 200,210,210
  200 HED1=PHI(I,J)
      HED2=KEEP(I,J)
      X=ABS(X)
      GO TO 220
  210 HED1=KEEP(I,J)
      HED2=PHI(I,J)
  220 STORE=S(I,J)
      IF (HED1-TOP(I,J).LE.0.) STORE=SY(I,J)
      IF ((HED1-TOP(I,J))*(HED2-TOP(I,J)).LT.0.0) STORE=(HED1-TOP(I,J))/
     1X*S(I,J)+(TOP(I,J)-HED2)/X*SY(I,J)
  230 STOR=STOR+STORE*(KEEP(I,J)-PHI(I,J))*AREA
C
C     ---COMPUTE LEAKAGE RATE---
      IF (LEAK.NE.CHK(9)) GO TO 240
      IF (M(I,J).EQ.0.) GO TO 240
      HED1=STRT(I,J)
      IF (CONVRT.EQ.CHK(7)) HED1=AMAX1(STRT(I,J),TOP(I,J))
      HED2=PHI(I,J)
      IF (CONVRT.EQ.CHK(7)) HED2=AMAX1(SNGL(PHI(I,J)),TOP(I,J))
      XX=RATE(I,J)*(RIVER(I,J)-HED1)*AREA/M(I,J)
      YY=TL(I,J)*(HED1-HED2)*AREA
      FLUX=FLUX+XX
      XNET=XX+YY
      FLUXS=FLUXS+XNET
      IF (XNET.LT.0.) FLXN=FLXN-XNET
  240 CONTINUE
C     ..................................................................
C
C     ---COMPUTE CUMULATIVE VOLUMES, TOTALS, AND DIFFERENCES---
      STORT=STORT+STOR
      STOR=STOR/DELT
      ETFLXT=ETFLXT-ETFLUX*DELT
      FLUXT=FLUXT+FLUXS*DELT
      FLXNT=FLXNT+FLXN*DELT
      FLXPT=FLUXT+FLXNT
      QRET=QRET+QREFLX*DELT
      CHDT=CHDT-CHD1*DELT
      CHST=CHST+CHD2*DELT
      PUMPT=PUMPT-PUMP*DELT
```

Program listing—Continued

```
      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 250
      PERCNT=DIFF/TOTL2*100.
  250 RETURN
C     ..................................................................
C
C     ---PRINT RESULTS---
C     *************************
      ENTRY CWRITE
C     *************************
C
      WRITE (P,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
      RETURN
C
C     ---FORMATS---
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
     $UM 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//)
      END


      SUBROUTINE PRNTAI(PHI,SURI,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
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
C
      DIMENSION PHI(IZ,JZ), SURI(IZ,JZ), S(IZ,JZ), WELL(IZ,JZ), DELX(JZ)
     1, DELY(IZ), T(IZ,JZ)
C
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LE
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DI
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2
```

Program listing—Continued

```
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(5),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
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      RETURN
C     ..................................................................
C
C     ---INITIALIZE VARIABLES FOR PLOT---
C     ********************
      ENTRY MAP
C     ********************
   10 XSF=DINCH*XSCALE
      YSF=DINCH*YSCALE
      NYD=YDIM/YSF
      IF (NYD*YSF.LE.YDIM-DELY(INO1)/2.) NYD=NYD+1
      IF (NYD.LE.12) GO TO 20
      DINCH=YDIM/(12.*YSCALE)
      WRITE (P,310) DINCH
      IF (YSCALE.LT.1.0) WRITE (P,320)
      GO TO 10
   20 NXD=WIDTH/XSF
      IF (NXD*XSF.LE.WIDTH-DELX(JNO1)/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
      DO 40 I=1,NE
      NNX=N5-I
      NNY=I-1
      IF (NNY.GE.N6) GO TO 30
      YN(I)=YSF*NNY/YSCALE
   30 IF (NNX.LT.0) GO TO 40
      XN(I)=XSF*NNX/YSCALE
   40 CONTINUE
      RETURN
C     ..................................................................
C
C     ********************
      ENTRY PRNTA(NG)
C     ********************
C     ---VARIABLES INITIALIZED EACH TIME A PLOT IS REQUESTED---
      DIST=WIDTH-DELX(JNO1)/2.
      JJ=JNO1
      LL=1
      Z=NXD*XSF
      IF (NG.EQ.1) WRITE (P,280) (TITLE(I),I=1,2)
      IF (NG.EQ.2) WRITE (P,280) (TITLE(I),I=3,5)
      DO 270 I=1,N4
C
C     ---LOCATE X AXES---
```

Program listing—Continued

```
      IF (I.EQ.1.OR.I.EQ.N4) GO TO 50                                   PRN 820
      PRNT(1)=SYM(12)                                                   PRN 830
      PRNT(N8)=SYM(12)                                                  PRN 840
      IF ((I-1)/N1*N1.NE.I-1) GO TO 70                                  PRN 850
      PRNT(1)=SYM(14)                                                   PRN 860
      PRNT(N8)=SYM(14)                                                  PRN 870
      GO TO 70                                                          PRN 880
C                                                                       PRN 890
C     ---LOCATE Y AXES---                                               PRN 900
   50 DO 60 J=1,N8                                                      PRN 910
      IF ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)                           PRN 920
   60 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
   70 IF (DIST.LT.0..OR.DIST.LT.Z-XN1*XSF) GO TO 220                    PRN 960
      YLEN=DELY(2)/2.                                                   PRN 970
      DO 200 L=2,INO1                                                   PRN 980
      J=YLEN*N2/YSF+1.5                                                 PRN 990
      IF (T(L,JJ).EQ.0.) GO TO 140                                      PRN1000
      IF (S(L,JJ).LT.0.) GO TO 190                                      PRN1010
      INDX3=0                                                           PRN1020
      GO TO (80,90), NG                                                 PRN1030
   80 K=(SURI(L,JJ)-PHI(L,JJ))*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 100                                                         PRN1070
   90 K=PHI(L,JJ)*FACT2                                                 PRN1080
  100 IF (K) 110,140,120                                                PRN1090
  110 IF (J-2.GT.0) PRNT(J-2)=SYM(13)                                   PRN1100
      N=-K                                                              PRN1110
      IF (N.LT.100) GO TO 130                                           PRN1120
      GO TO 170                                                         PRN1130
  120 N=K                                                               PRN1140
      IF (N.LT.100) GO TO 130                                           PRN1150
      IF (N.GT.999) GO TO 170                                           PRN1160
      INDX3=N/100                                                       PRN1170
      IF (J-2.GT.0) PRNT(J-2)=SYM(INDX3)                                PRN1180
      N=N-INDX3*100                                                     PRN1190
  130 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 150                                            PRN1230
      INDX2=N/10                                                        PRN1240
      IF (INDX2.GT.0) GO TO 160                                         PRN1250
      INDX2=10                                                          PRN1260
      IF (INDX3.EQ.0) INDX2=15                                          PRN1270
      GO TO 160                                                         PRN1280
  140 INDX1=15                                                          PRN1290
  150 INDX2=15                                                          PRN1300
  160 IF (J-1.GT.0) PRNT(J-1)=SYM(INDX2)                                PRN1310
      PRNT(J)=SYM(INDX1)                                                PRN1320
      GO TO 200                                                         PRN1330
  170 DO 180 II=1,3                                                     PRN1340
      JI=J-3+II                                                         PRN1350
  180 IF (JI.GT.0) PRNT(JI)=SYM(11)                                     PRN1360
  190 IF (S(L,JJ).LT.0.) PRNT(J)=SYM(16)                                PRN1370
  200 YLEN=YLEN+(DELY(L)+DELY(L+1))/2.                                  PRN1380
  210 DIST=DIST-(DELX(JJ)+DELX(JJ-1))/2.                                PRN1390
      JJ=JJ-1                                                           PRN1400
      IF (JJ.EQ.0) GO TO 220                                            PRN1410
      IF (DIST.GT.Z-XN1*XSF) GO TO 210                                  PRN1420
```

Program listing—Continued

```
  220 CONTINUE
C
C     ---PRINT AXES,LABELS, AND SYMBOLS---
      IF (I-NA(LL).EQ.0) GO TO 240
      IF ((I-1)/N1*N1-(I-1)) 250,230,250
  230 WRITE (P,VF1) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XN(1+(I-1)/6)
      GO TO 260
  240 WRITE (P,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XLABEL(LL)
      LL=LL+1
      GO TO 260
  250 WRITE (P,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8)
C
C     ---COMPUTE NEW VALUE FOR Z AND INITIALIZE PRNT---
  260 Z=Z-2.*XN1*XSF
      DO 270 J=1,N8
  270 PRNT(J)=SYM(15)
C
C     ---NUMBER AND LABEL Y AXIS AND PRINT LEGEND---
      WRITE (P,VF3) (BLANK(J),J=1,NC),(YN(I),I=1,N6)
      WRITE (P,300) (YLABEL(I),I=1,6)
      IF (NG.EQ.1) WRITE (P,290) FACT1
      IF (NG.EQ.2) WRITE (P,290) FACT2
      RETURN
C
C     ---FORMATS---
C
C     ------------------------------------------------------------------
C
C
  280 FORMAT ('1',53X,4A8//)
  290 FORMAT ('0EXPLANATION'/' ',11('-')//' R = CONSTANT HEAD BOUNDARY'/
     1' *** = VALUE EXCEEDED 3 FIGURES'/' MULTIPLICATION FACTOR =',F8.3)
  300 FORMAT ('0',39X,6A8)
  310 FORMAT ('0',25X,10('*'),' TO FIT MAP WITHIN 12 INCHES, DINCH REVIS
     1ED TO',G15.7,1X,10('*'))
  320 FORMAT ('0',45X,'NOTE: GENERALLY SCALE SHOULD BE > OR = 1.0')
      END
```

```
      BLOCK DATA
C     ----------
      REAL *8XLABEL,YLABEL,TITLE,XN1,MESUR,RHO,B,D,F,H
      INTEGER R,P,PU,DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
C
      COMMON /DPARAM/ RHO,B,D,F,H
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /SPARAM/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,EROR,LE
     1AK,RECH,SIP,U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,WIDTH,
     2NUMS,LSOR,ADI,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,
     3IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,ITMAX,LENGTH,NWEL,NW,DIML,DI
     4MW,JNO1,INO1,R,P,PU,I,J,IDK1,IDK2
      COMMON /PR/ XLABEL(3),YLABEL(6),TITLE(5),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
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
C     ******************************************************************
C
      DATA IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IMAX/13*20/,IH/1/
      DATA CHK/'PUNC','WATE','CONT','NUME','CHEC','EVAP','CONV','HEAD','
```

Program listing—Continued

```
     1LEAK','RECH','SIP ','LSOR','ADI','DK1 ','DK2 '/,R,P,PU/5,6,7/,B,D,
     2F,H/4*0.D0/
      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)'/
      DATA VF4/'(1H0',' ','    ','X,I2',',2X,','20F6','.1/(','    ','X,2
     10','F6.1','))'/
C     ******************************************************************
      END
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980 O— 311-344/13