USGS
science for a changing world

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

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 $\beta' \leq 0.6$ (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 ω_{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 ω_{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.

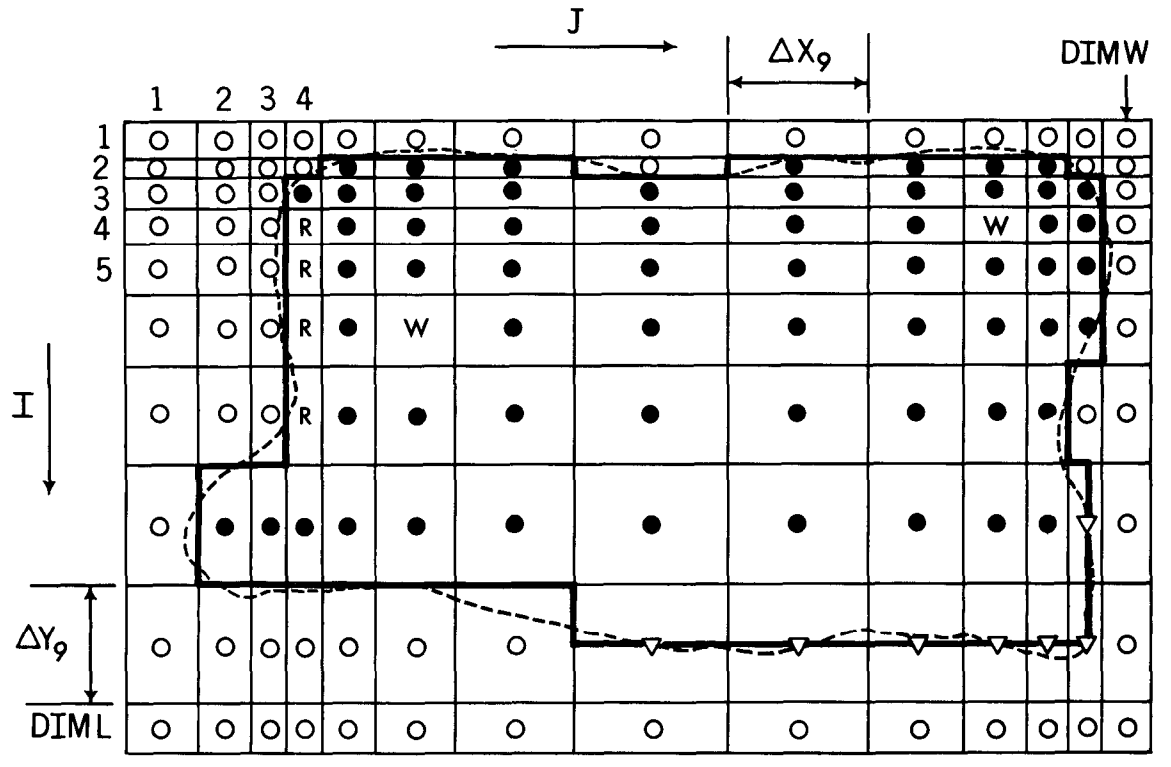
If initial conditions are specified so that transient flow is occurring in the system at the start of the simulation, it should be recognized that water levels will change during the simulation, not only in response to the new pumping stress, but also due to the initial conditions. This may or may not 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,



EXPLANATION

Node symbols

Inside aquifer (transmissivity >0)

- w Discharge well
- R Recharge well
- ▽ Constant head
- ● Node without wells or specified head

Outside aquifer

- o Transmissivity = 0

Aquifer boundary

Mathematical boundary

DIML Number of rows

DIMW Number of columns

Boundary conditions

Constant head

Constant flux

$\frac{\partial h}{\partial x} = 0$

R $\frac{\partial h}{\partial x} = C$

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 $(\overline{\overline{A}+\overline{B}})$;
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_s	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 $(\overline{\overline{A}+\overline{B}})$;
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});
$\dot{q}$	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 $(\overline{\overline{A}+\overline{B}})$;
$\overline{V}$	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 $(\overline{\overline{A}+\overline{B}})$;
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);

$\bar{\zeta}$	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_i	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

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

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

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

$DELT_o$ is the time step recorded on the data card.

The program has two options for selecting the time parameters:

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

NUMT as specified on the time parameter card.

To minimize the error due to approximation of the time derivative, several time steps should be simulated before the first step at which results are displayed. This suggestion should be followed unless the system is nearly 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-

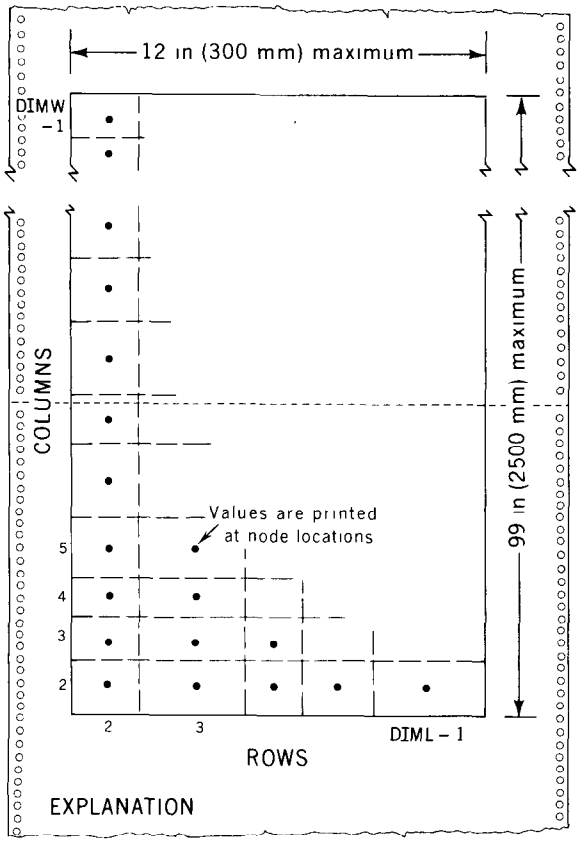


FIGURE 26.—Orientation of map on computer page.

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(≦6440)
eeee—equals DIMLxDIMWx8+52(≦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.

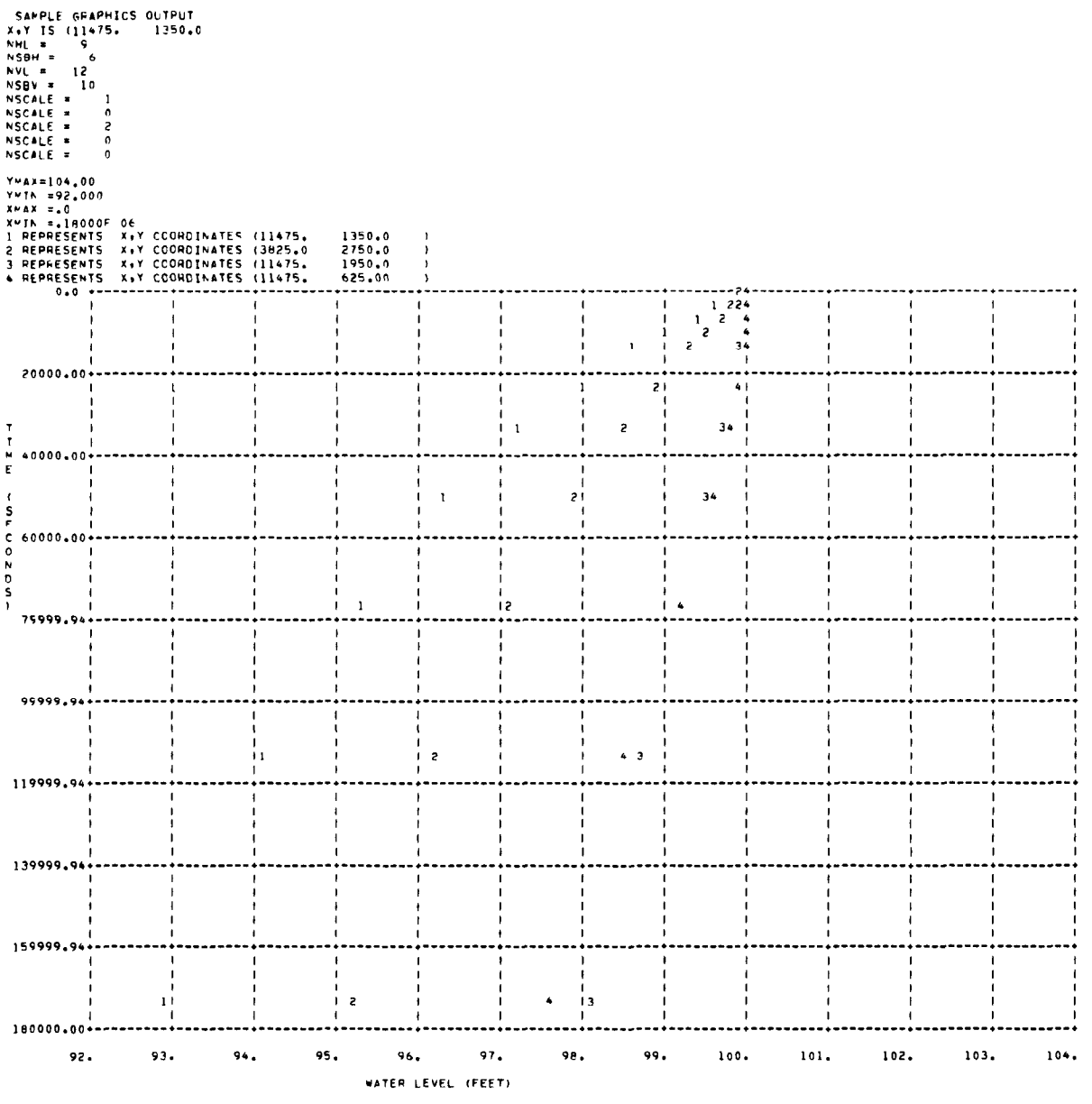


FIGURE 28.—Water level versus time at various nodes of the sample aquifer problem produced by the graphical display package.

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.

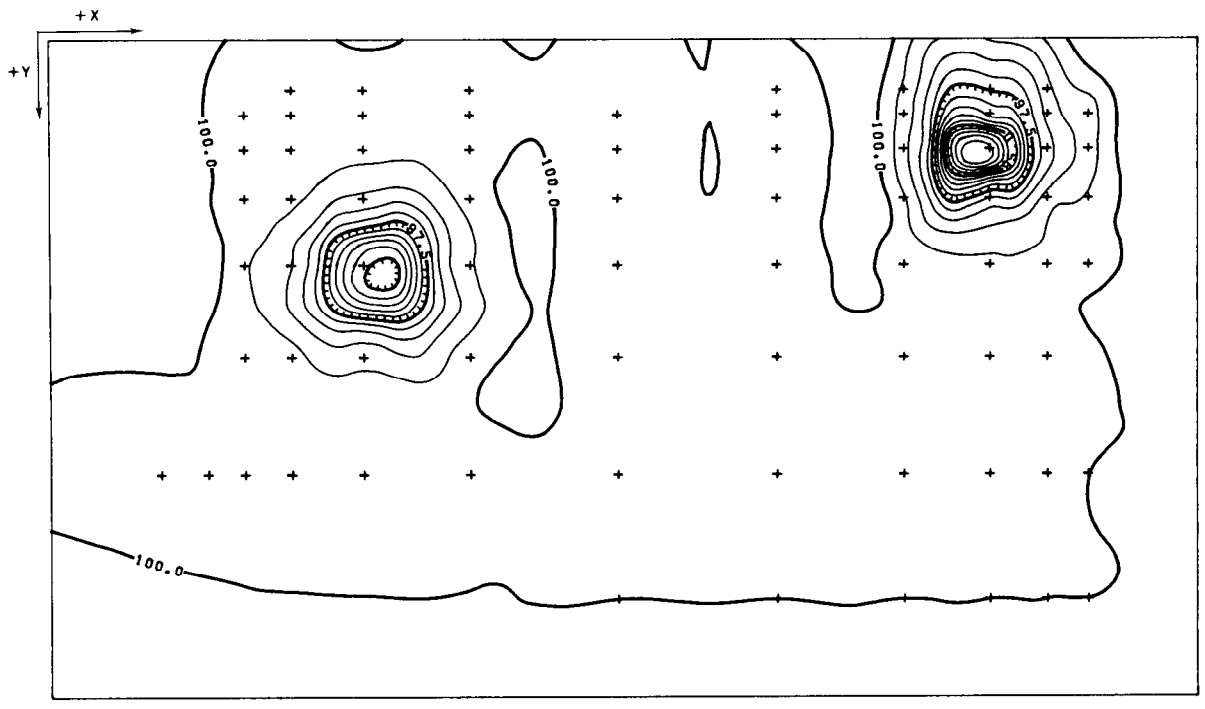


FIGURE 29.—Contour map of water level (in feet) for sample aquifer problem produced by graphical display package. Contour interval is 0.5 ft.

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.		
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.		
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.		
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.		
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 $\leq$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	I	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.

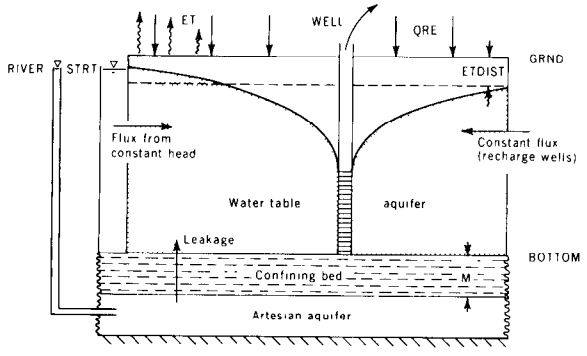


FIGURE 30.—Cross section illustrates several options included in the sample problem and identifies the meaning of several program parameters.

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-