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

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

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

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

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

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)

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

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

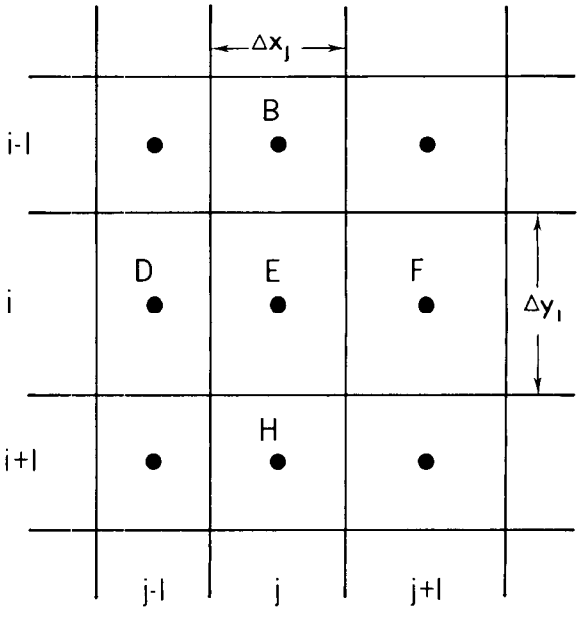


FIGURE 1.—Index scheme for finite-difference grid and coefficients of finite-difference equation written for node (i, j).

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

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

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

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

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

in which

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

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,

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

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

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

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

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

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

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

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});

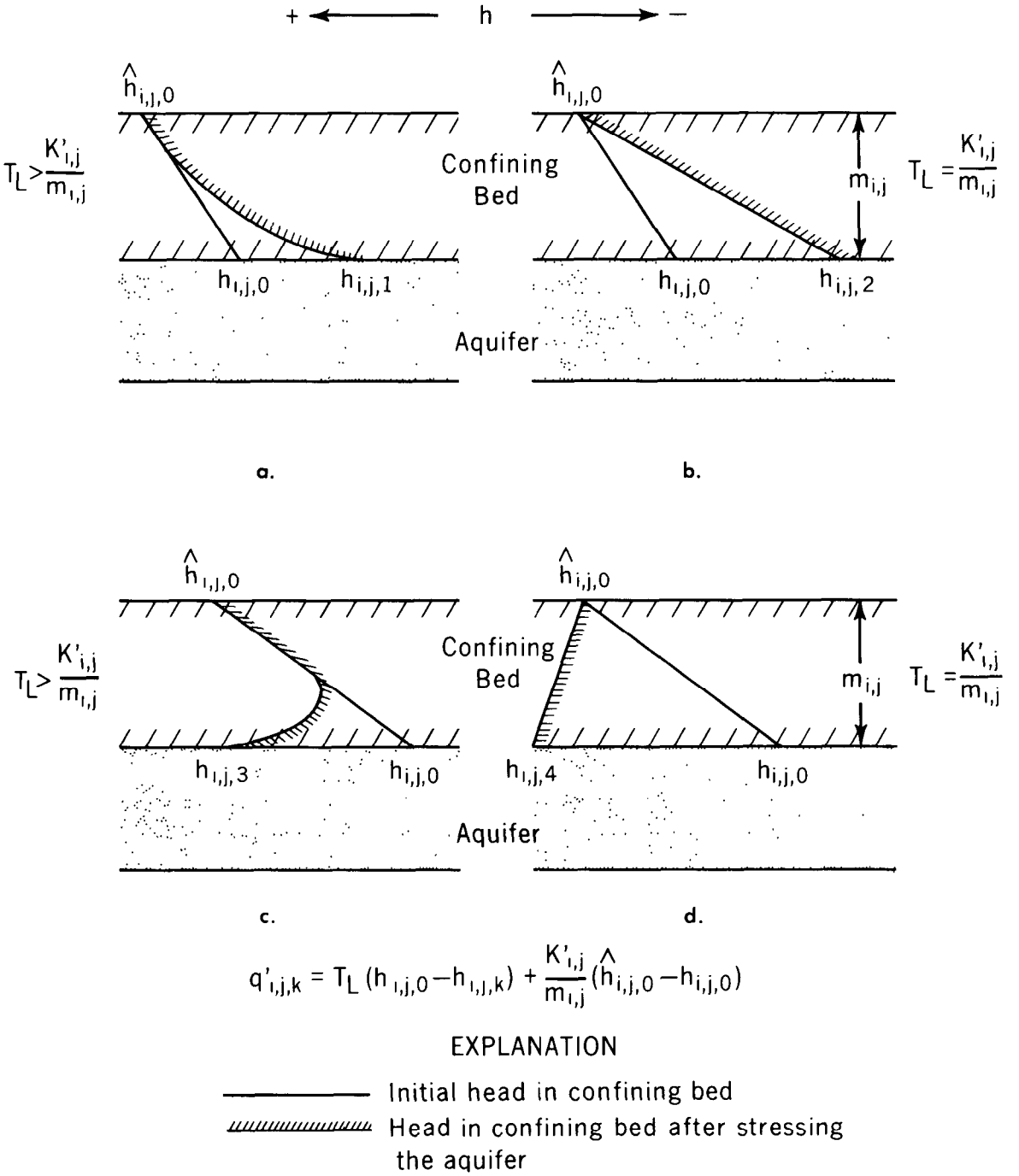
$(K'_{i,j}t/m^2_{i,j}S_{s[i,j]})$ is dimensionless time; see Bredehoeft and Pinder (1970) for a discussion of leakage versus dimensionless time;

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 $h_{i,j,1}$, . . . $h_{i,j,4}$.) 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 low-

ered 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



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

EXPLANATION

Initial head in confining bed

Head in confining bed after stressing the aquifer

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

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

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

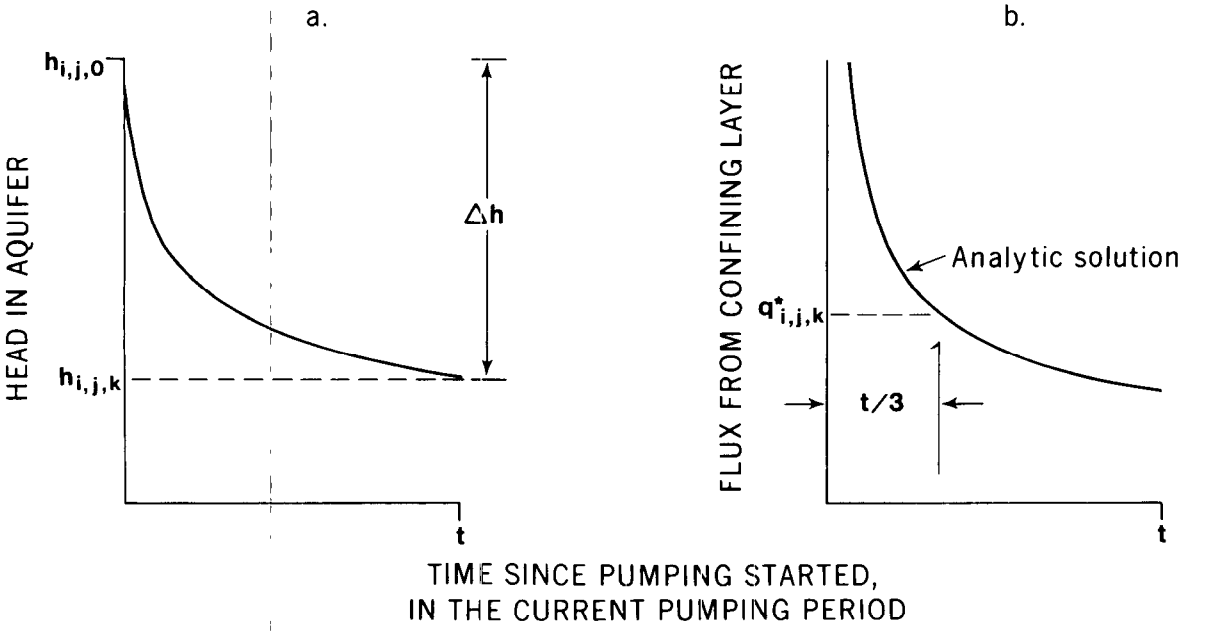


FIGURE 3.—The total drawdown at the elapsed time, t, in the pumping period (a) is applied at $t/3$ in equations 9 and 10 to approximate $q^*_{i,j,k}$, the transient part of $q'_{i,j,k}$ (b).

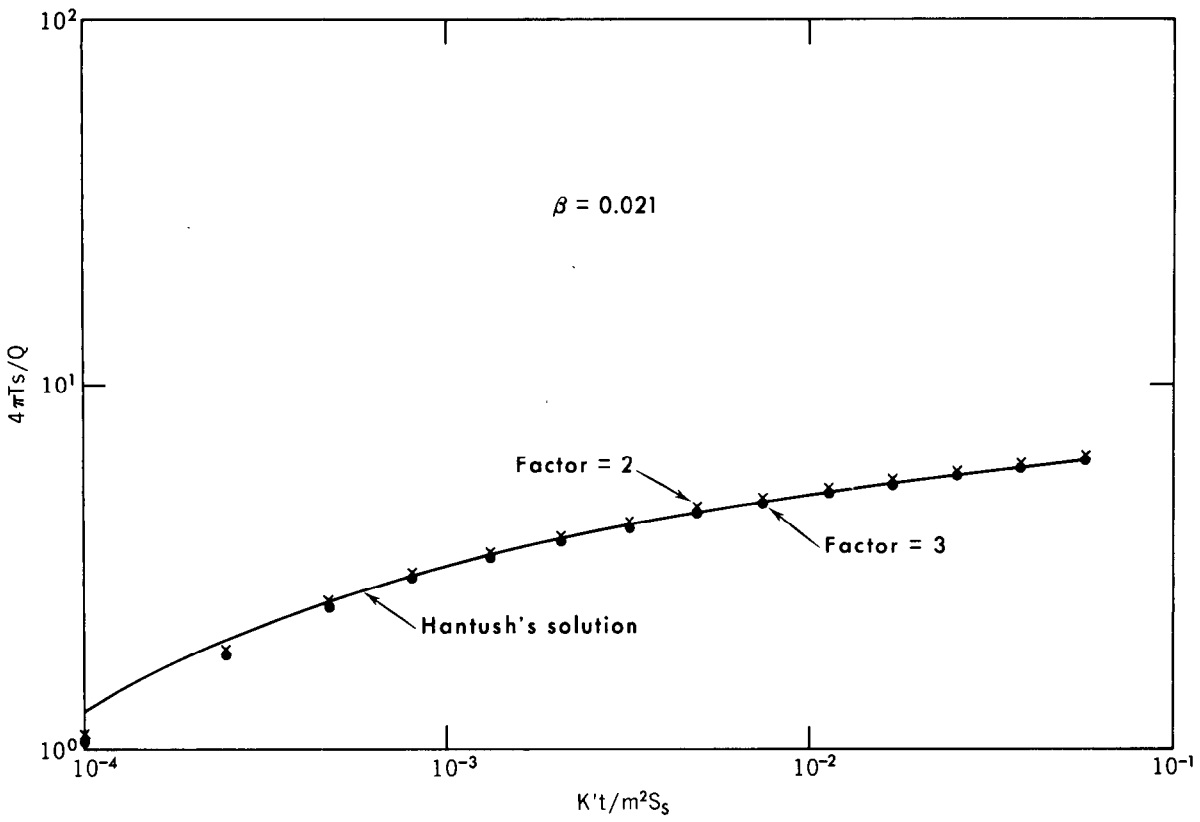


FIGURE 4.—Comparison of analytic solution and numerical results using factors of 2 and 3 in the transient leakage approximation.

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

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

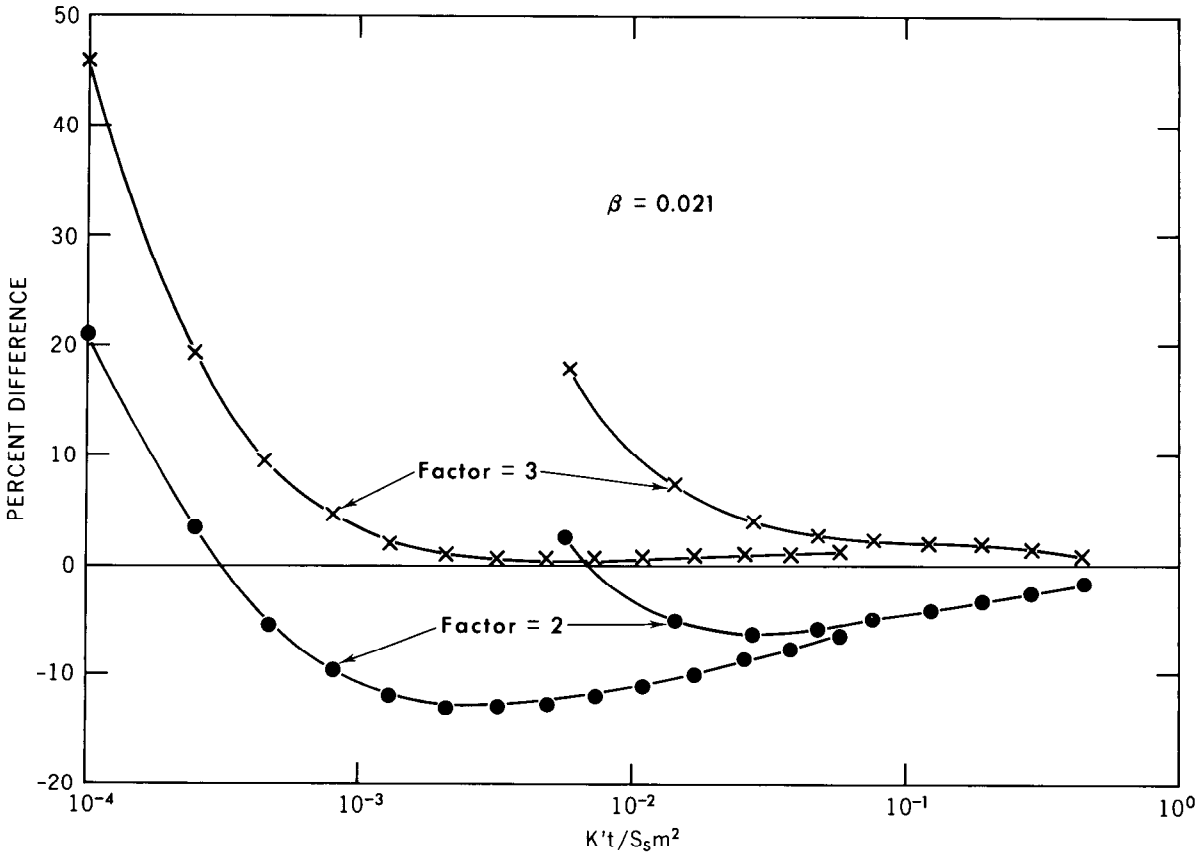


FIGURE 5.—Percent difference between the volume of leakage computed with the model approximation and Hantush's analytical results.

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).

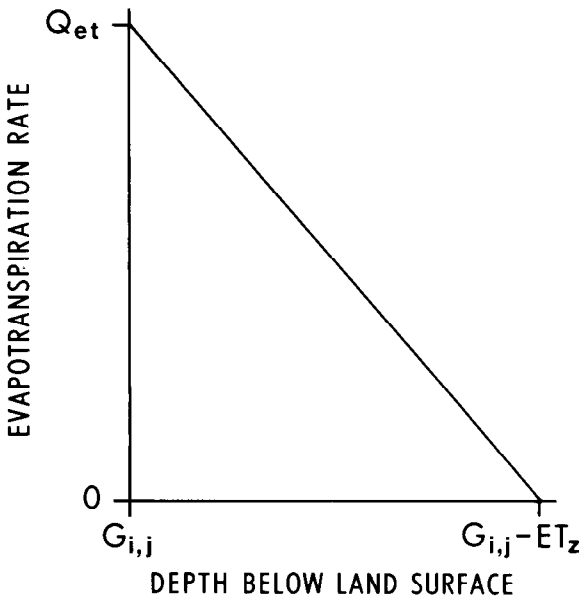


FIGURE 6.—Evapotranspiration decreases linearly from Q_{et} where the water table is at land surface to zero where the water table is less than or equal to $G_{i,j}-ET_z$.

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)

$$r_e = r_1/4.81 \qquad (12)$$

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

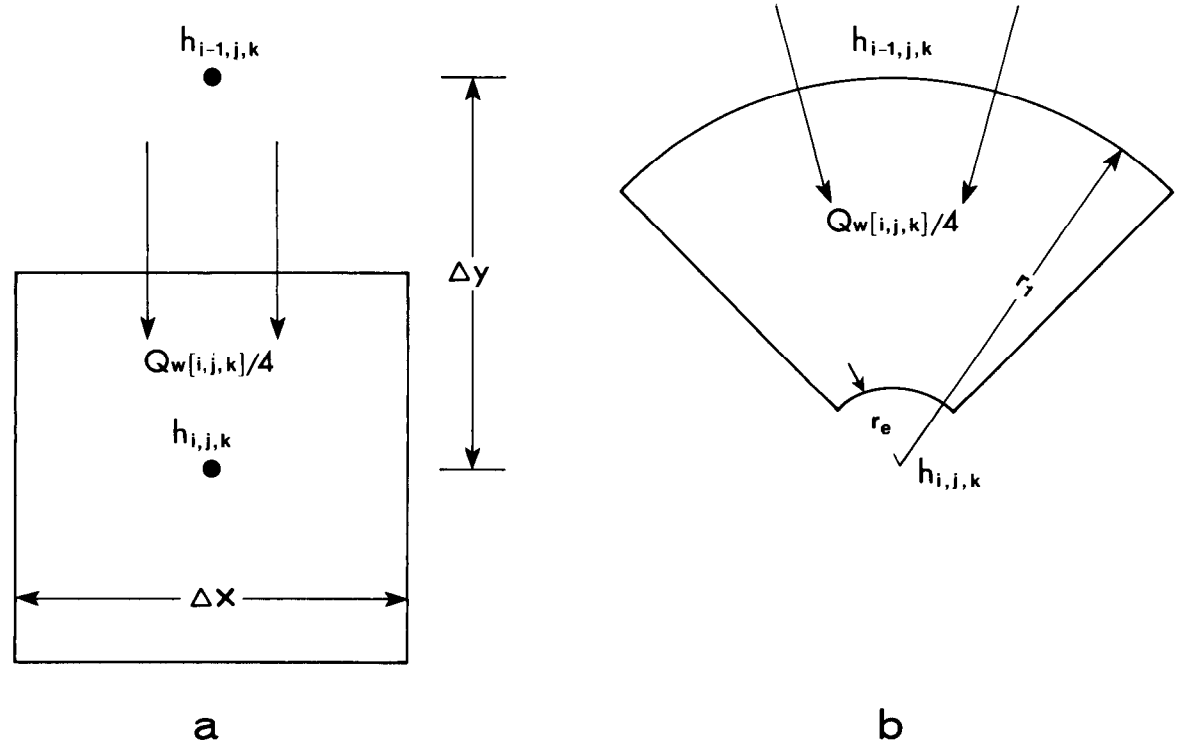


FIGURE 7.—Flow from cell $(i-1,j)$ to cell (i,j)(a) and equivalent radial flow to well (i,j) with radius r_e(b).

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

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

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

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 artesian		Tt/r^2S		
	3	3.0×10^5	41.1	42.7
	14	3.7×10^7	58.3	58.1
Leaky artesian		$K't/m^2S_s$		
	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

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

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

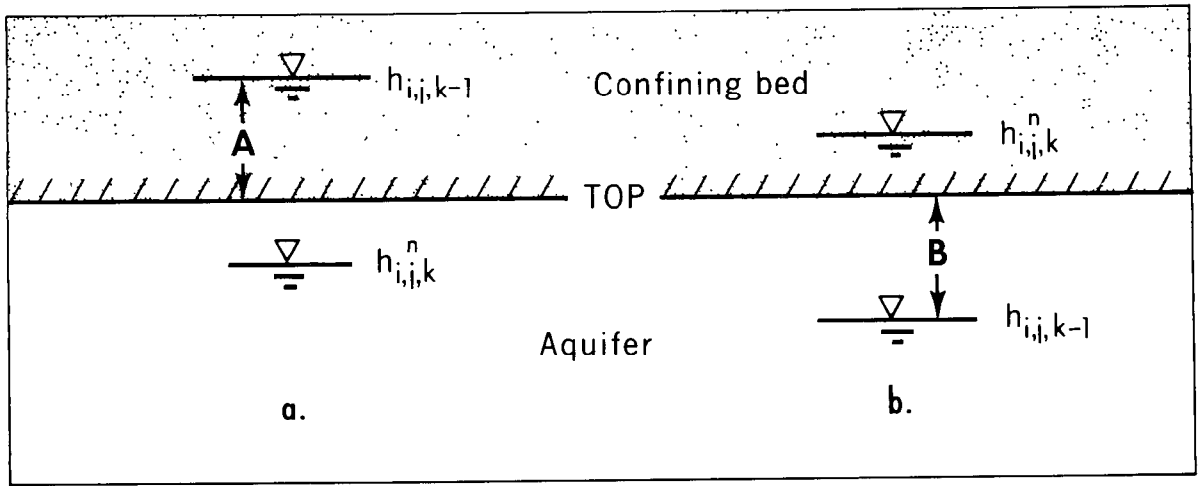


FIGURE 8.—Storage adjustment is applied to distance A in conversion from artesian to water-table conditions (a) and to distance B in conversion from water-table to artesian conditions (b).

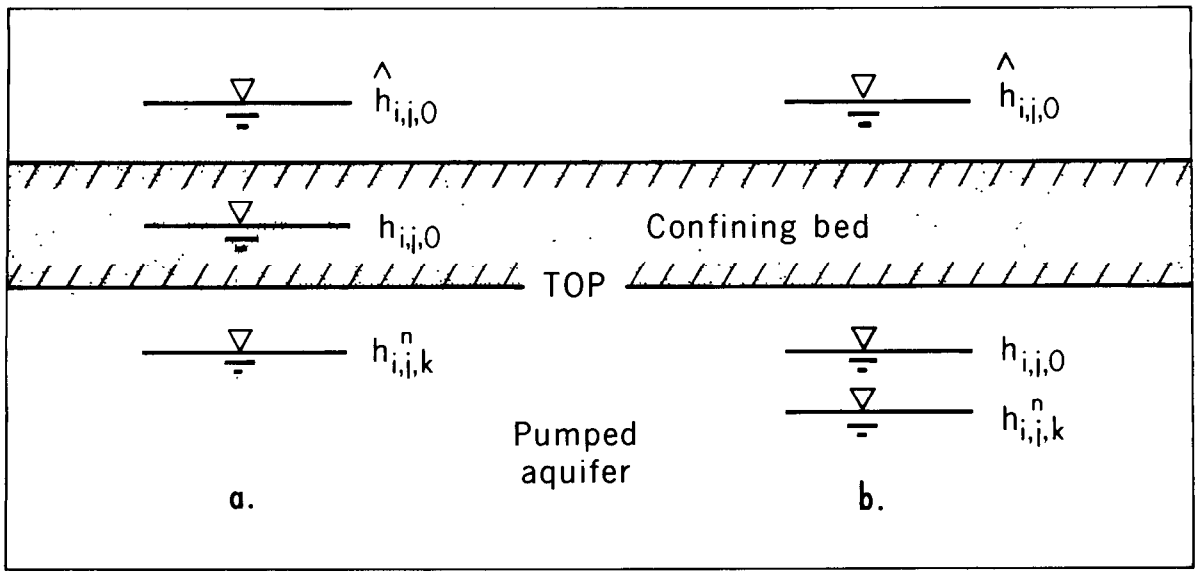


FIGURE 9.—Two of the possible situations in which leakage is restricted in artesian–water-table simulations.

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

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:

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

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 ft^2/s (0.009m^2/s)
Grid spacing: $\Delta x = \Delta y$ = 5000 ft (1500m)
Dimensions of grid. 18×18

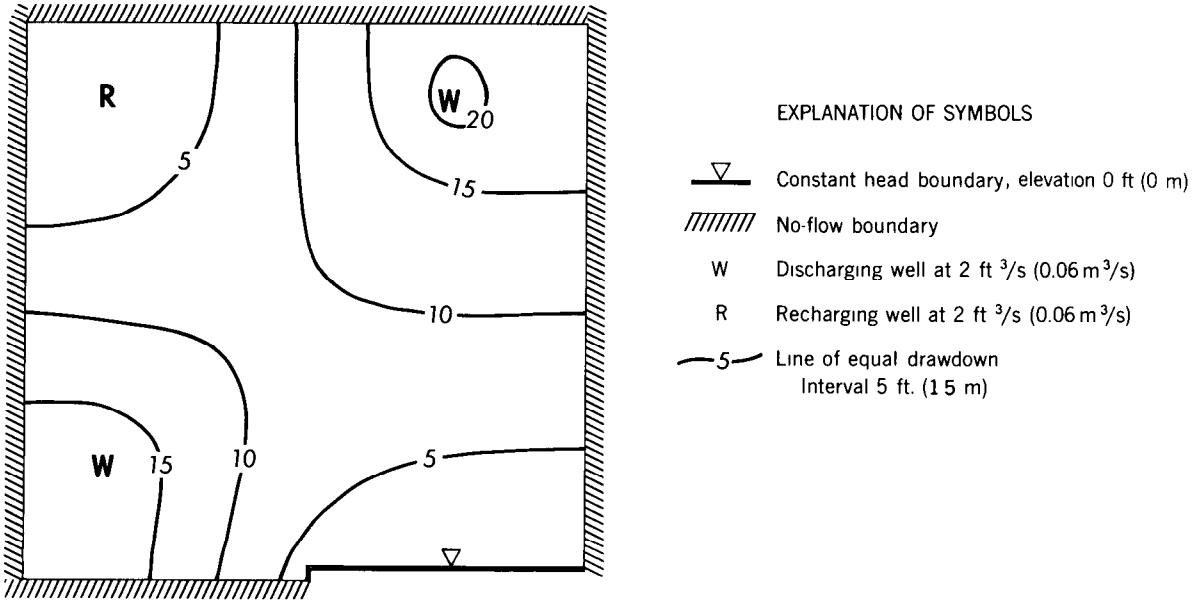


FIGURE 10.—Characteristics of test problem 1.

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

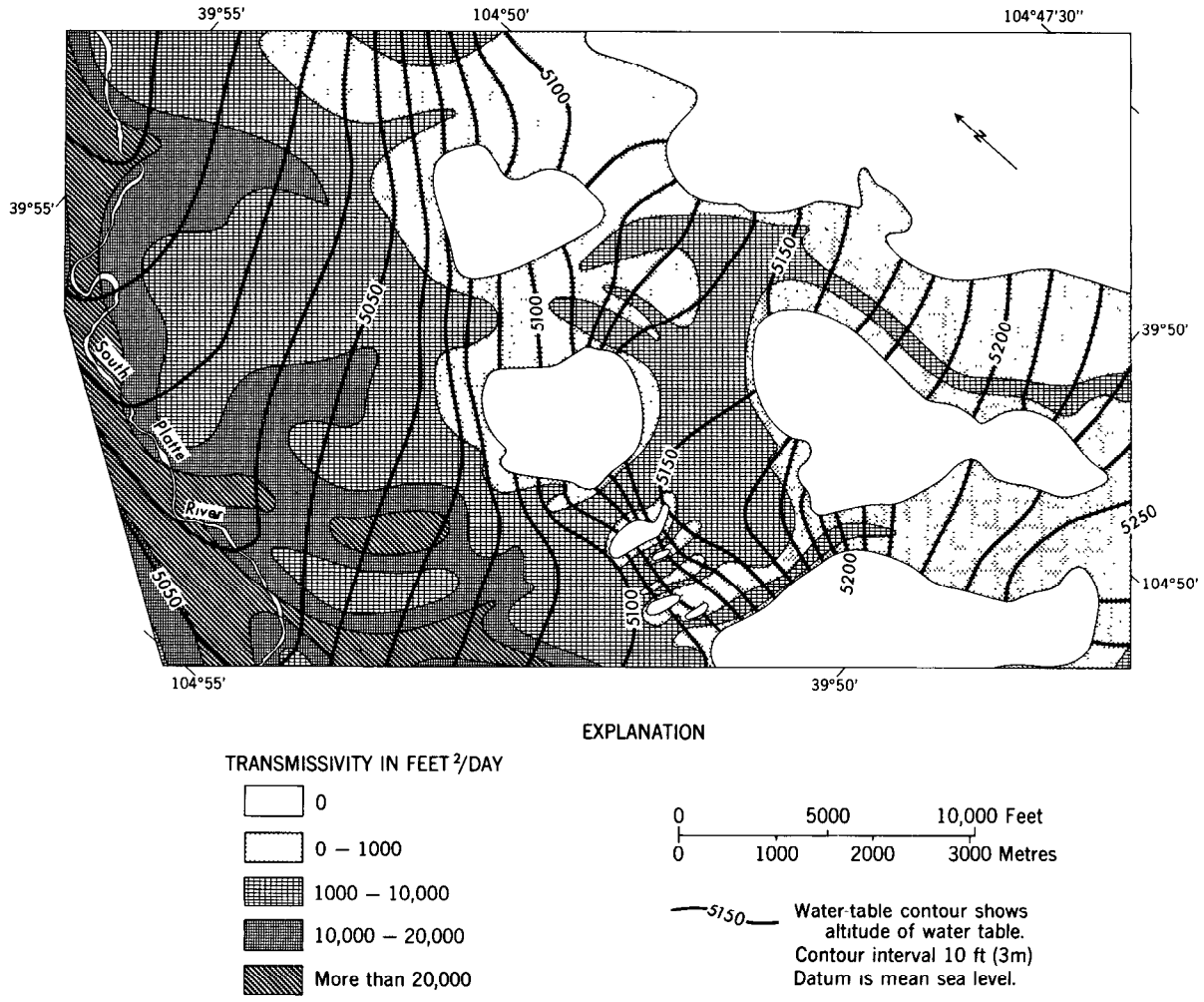


FIGURE 11.—Transmissivity and observed water-table configuration for test problem 2 (fieldwork and model design by Konikow, 1975).

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,

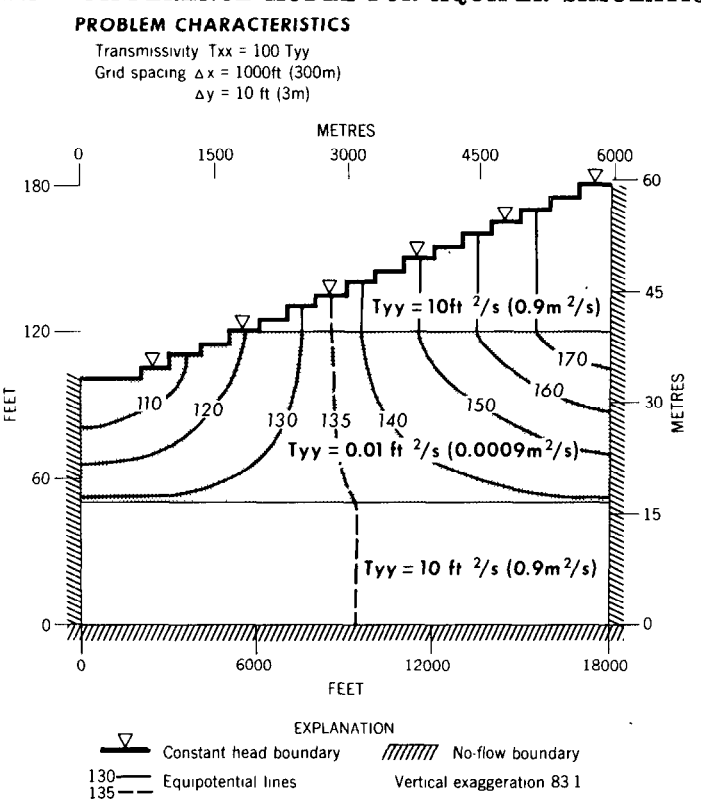


FIGURE 12.—Characteristics of test problem 3.

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):

row 1

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

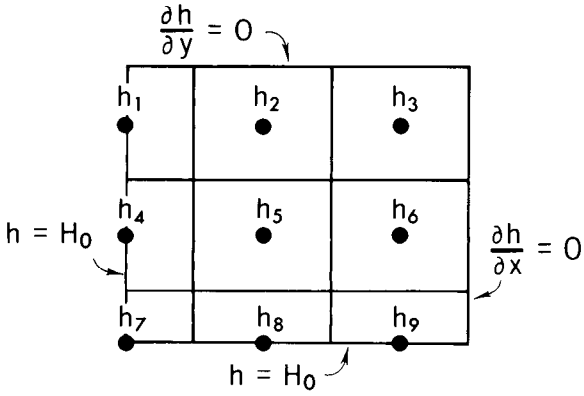


FIGURE 13.—Hypothetical problem with 9 interior nodes.

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

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

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

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

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 $\hat{\beta}$ 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

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

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 $\bar{\hat{\beta}}$ 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

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

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.)

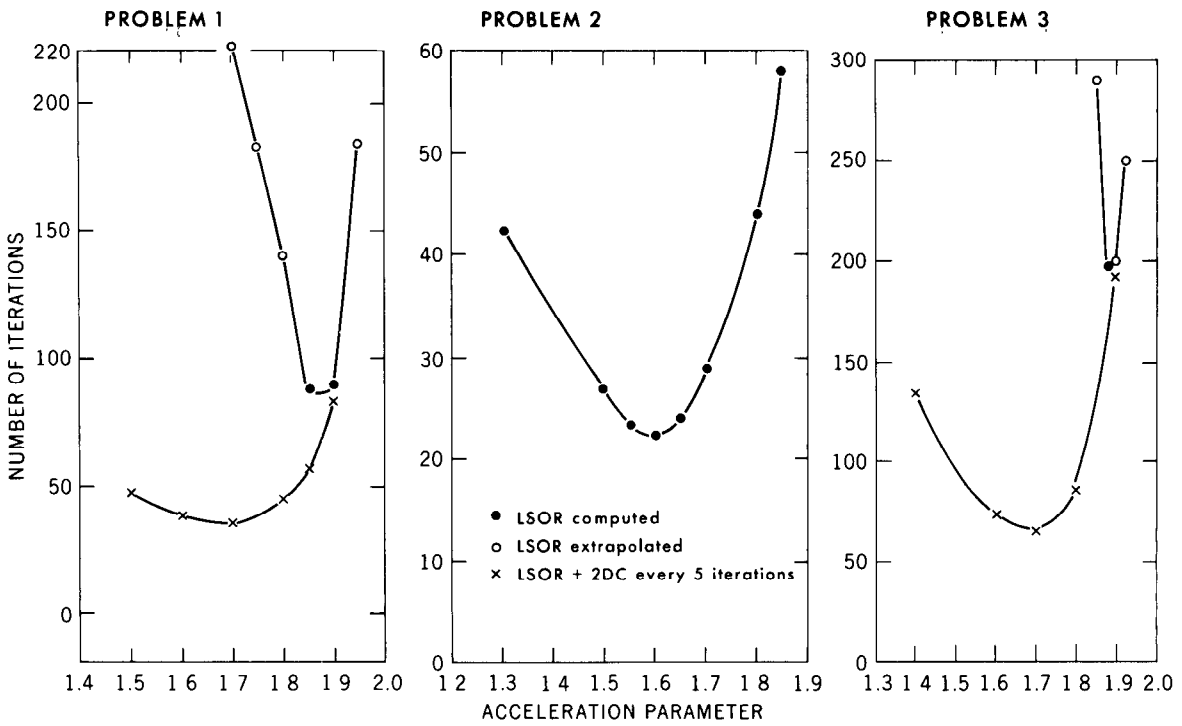


FIGURE 14.—Number of iterations required for solution by LSOR and LSOR + 2DC using different acceleration parameters.

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

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

in which

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

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

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:

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

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

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

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

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

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

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

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

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

in which

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

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 ex-

cessive 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

$$\bar{\bar{A}}\,\bar{h}=\bar{Q} \tag{24}$$

N_x+1 elements; N_x+1 elements

$$\begin{bmatrix}
E_1 & F_1 & 0 & H_1 & & & & & \\
D_2 & E_2 & F_2 & 0 & H_2 & & & & \\
0 & D_3 & E_3 & 0 & 0 & H_3 & & & \\
B_4 & 0 & 0 & E_4 & F_4 & 0 & H_4 & & \\
 & B_5 & 0 & D_5 & E_5 & F_5 & 0 & H_5 & \\
 & & B_6 & 0 & D_6 & E_6 & 0 & 0 & H_6 \\
 & & & B_7 & 0 & 0 & E_7 & F_7 & 0 \\
 & & & & B_8 & 0 & D_8 & E_8 & F_8 \\
 & & & & & B_9 & 0 & D_9 & E_9
\end{bmatrix}
\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix}
=
\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ Q \\ Q_8 \\ Q_9 \end{bmatrix}$$

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

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

$(\overline{\overline{A+B}})$ 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

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

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

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

in which

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

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

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

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);

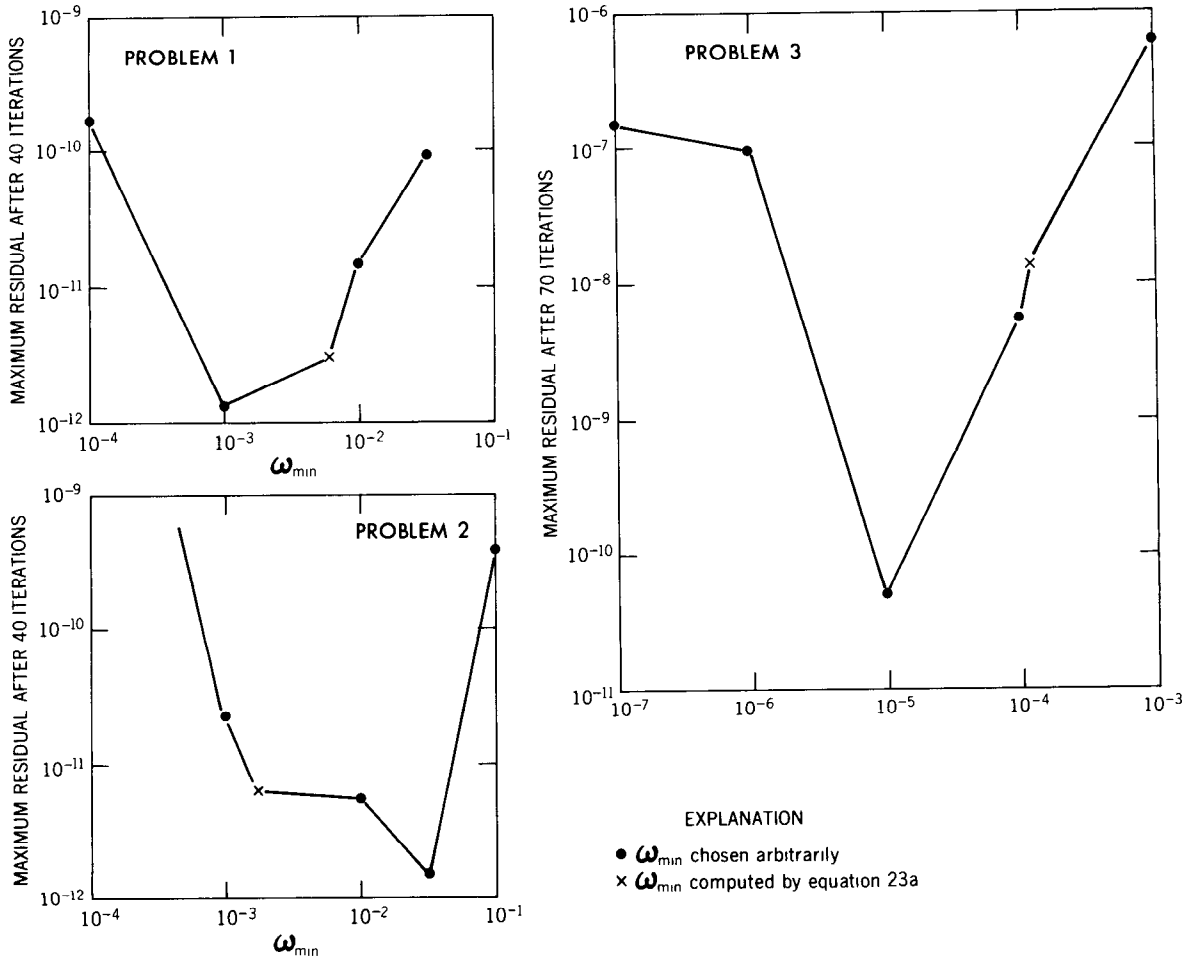


FIGURE 15.—Reduction in the maximum residual for problems 1 to 3 for selected ω_{min} used to compute the ADI parameters.

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

(N_x Elements)

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

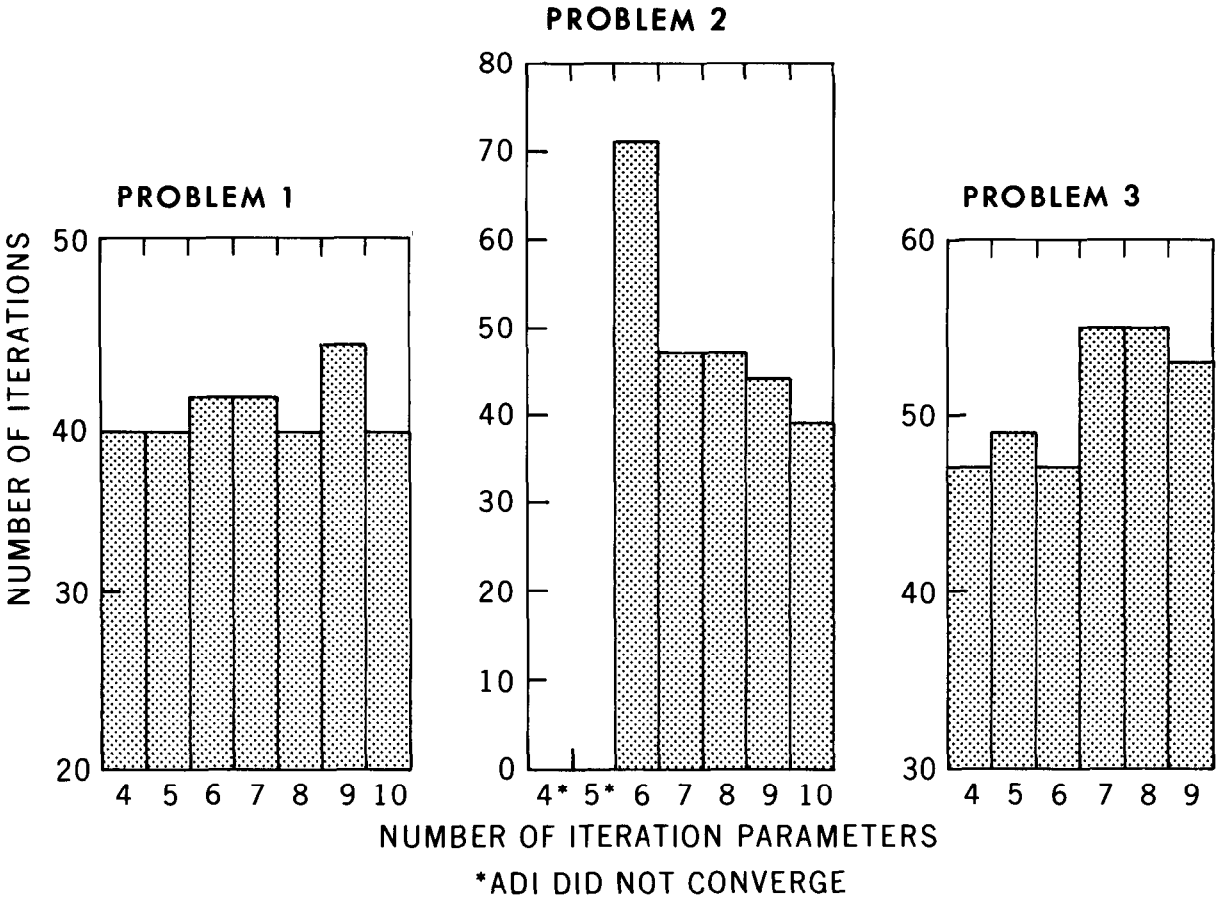


FIGURE 16.—Number of iterations required for solution of the test problems with ADI using different numbers of parameters.

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

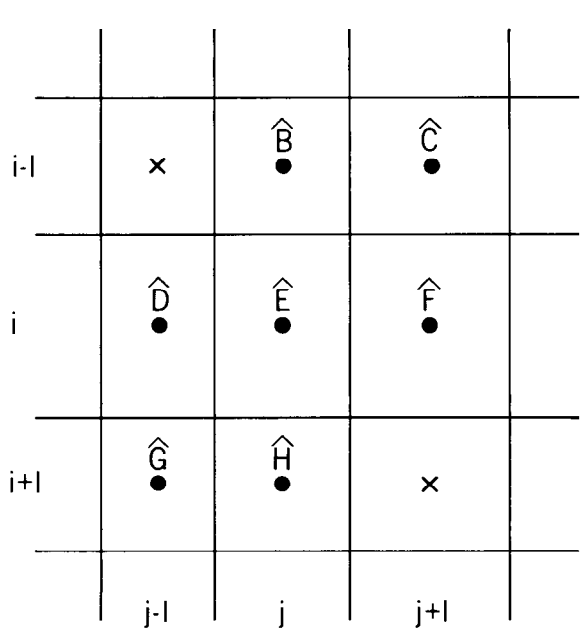


FIGURE 17.—Coefficients of unknowns in equation 27.

$\xi_{i+1,j-1}$ 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.

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

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

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

$$\bar{\bar{L}}\bar{\bar{U}}\bar{\xi}^n=\bar{R}^{n-1} \qquad (35)$$

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

$$\bar{\bar{U}}\bar{\xi}^n=\bar{V}^n. \qquad (36)$$

Then equation 35 becomes

$$\bar{\bar{L}}\bar{V}^n=\bar{R}^{n-1}. \qquad (37)$$

$\bar{V}^n$ 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

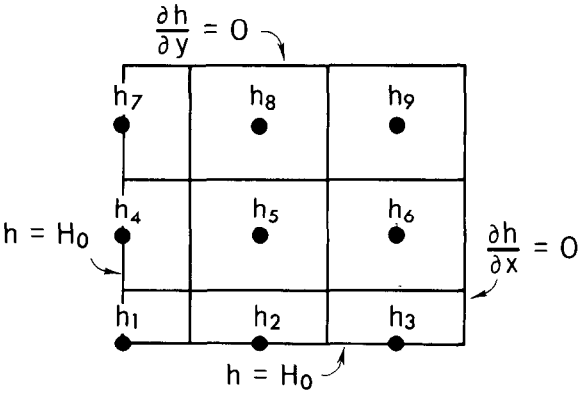


FIGURE 18.—Reverse numbering scheme for 3× 3 problem.

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

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

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

in which

$\delta x=\Delta x_j/\text{width of model}$

$\delta y=\Delta y_i/\text{length of model}$

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.

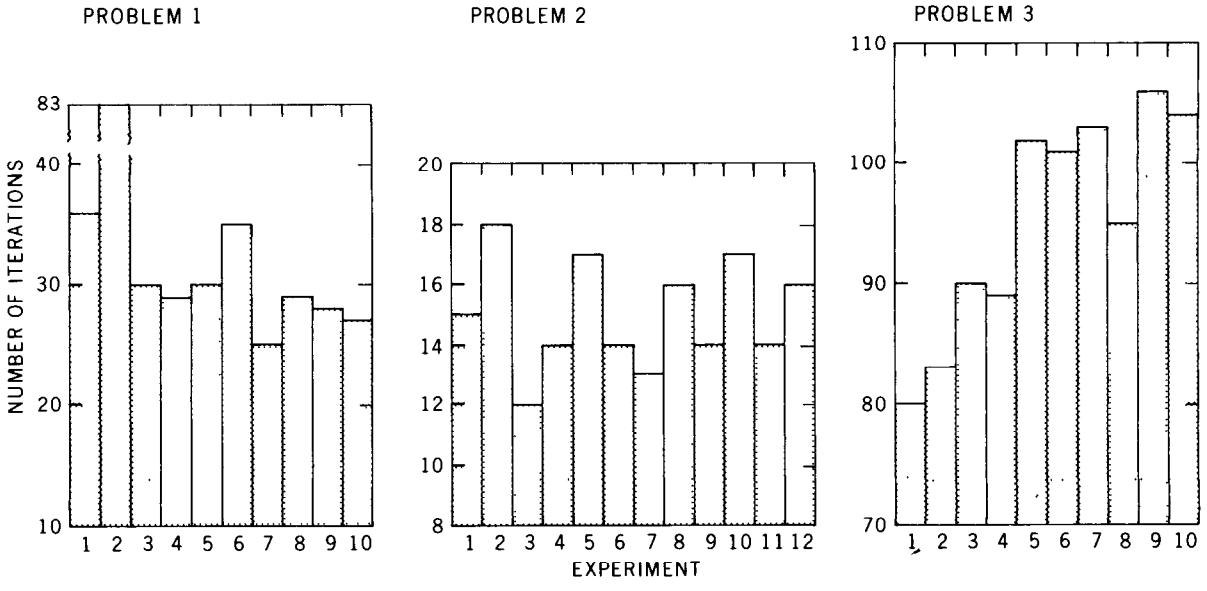
Stone (1968) recommends using a minimum of four parameters, each used twice in

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

Experiment	Sequence of parameters
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.

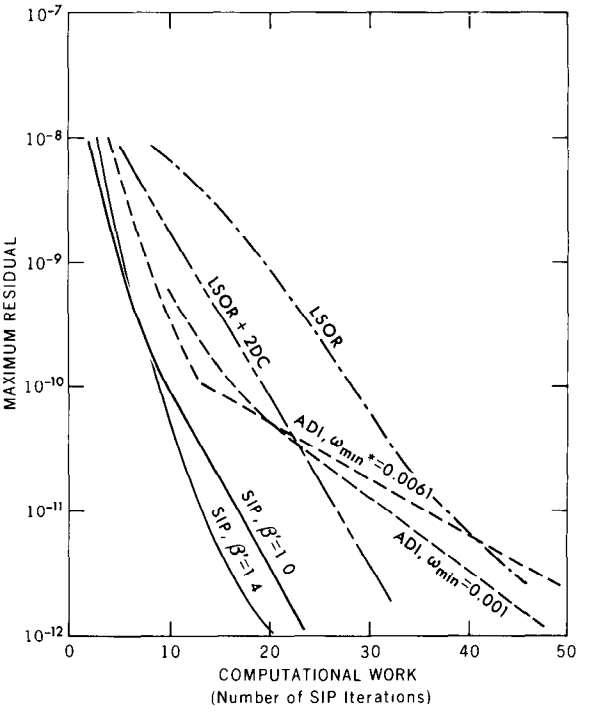


FIGURE 20.—Computational work required by different iterative techniques for problem 1.

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.

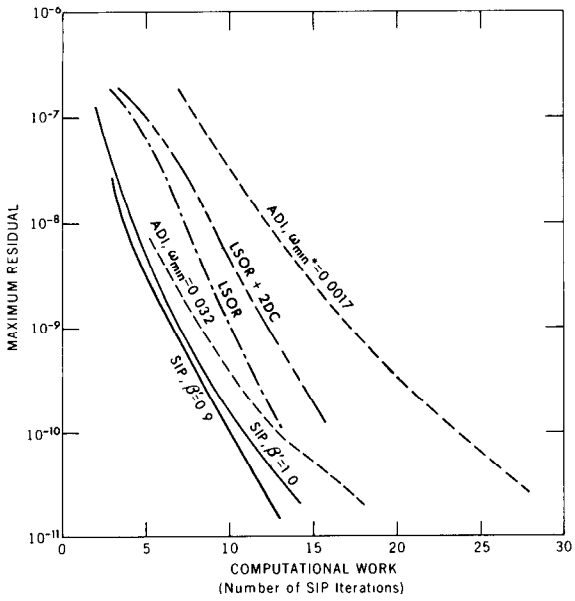


FIGURE 21.—Computational work required by different iterative techniques for problem 2.

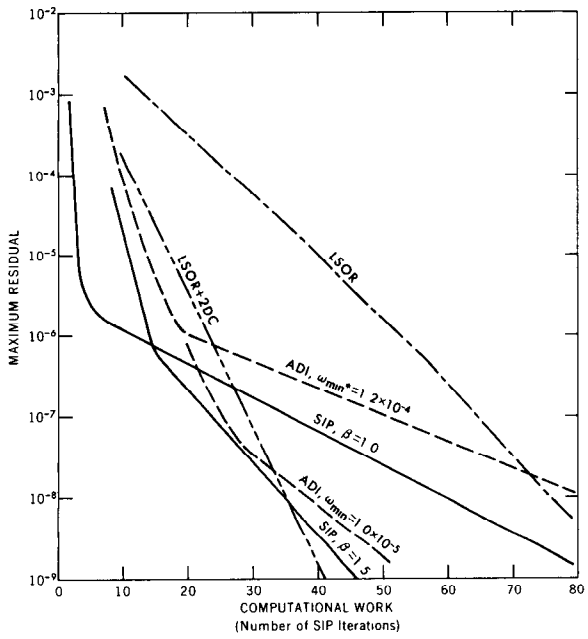


FIGURE 22.—Computational work required by different iterative techniques for problem 3.

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

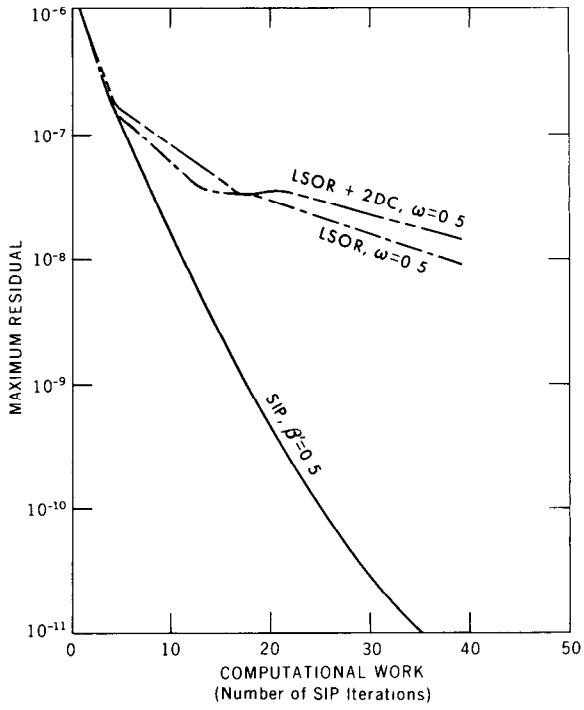


FIGURE 23.—Computational work required by different iterative techniques for problem 4.

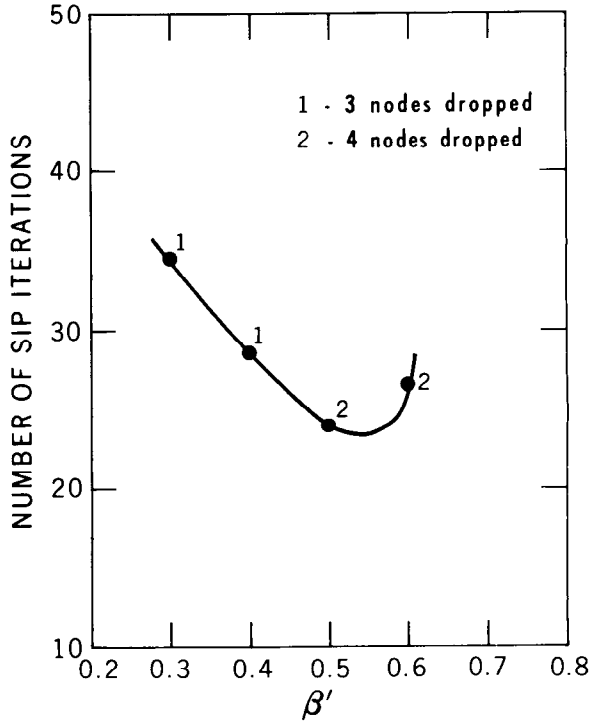


FIGURE 24.—Number of iterations required for solution of problem 4 by SIP using different values of β'.

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 $\beta' \leq 0.6$ (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.