(200)

UNITED STATES
DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

A PROGRAM TO COMPUTE AQUIFER-RESPONSE COEFFICIENTS

Open-File Report 75-612

A PROGRAM TO COMPUTE AQUIFER-RESPONSE COEFFICIENTS

BY

THOMAS MADDOCK, III

ABSTRACT

An alternating direction technique is used to solve finite difference equations approximating the flow of water in an aquifer. The solutions produce response coefficients relating pumping from wells to drawdowns within those wells. The product of the response coefficient with the pumping values produces a linear algebraic technological function that can be used for integrating hydrologic phenomena into planning and management models.

TABLE OF CONTENTS

APPENDICES

ILLUSTRATIONS

Introduction

This program, written in FORTRAN IV for the IBM 360 or 370 series computers, calculates the response coefficients for one or more wells over a set of specified time periods called the design horizon. For convenience the time period selected represents the length of time during which the rate of pumping of a well remains constant. The program may simulate 1) a confined aquifer or 2) an unconfined aquifer providing the drawdown is small in relation to the saturated thickness (transmissivity is thus independent of the drawdown). The aquifer may have irregular shaped boundaries and non-homogeneous transmissivity. Vertical components of flow in the aquifer should be neglible compared to the horizontal components of flow. The response functions calculated for the wells are used to determine drawdown averaged over the area represented by a node. No well bore correction term is present in this version of the program.

Application

This program may be used to calculate response coefficients relating pumping to drawdown. The response coefficients are valid only if the following characteristic behavior and restrictions are followed.

1. The region of interest is underlain by a single aquifer.
2. The aquifer may be treated as a two-dimensional flow system.
3. Drawdowns relative to the saturated thickness are always small (transmissivity is essentially independent of drawdown).
4. The storage coefficient throughout the aquifer may be treated as a constant.
5. There is no land subsidence and water is instantly released from storage.

6. The boundaries may be regular or irregular in shape.

7. The transmissivity may be nonhomogeneous but isotropic.

8. The aquifer may have constant head boundaries.

9. Time is broken up into equal intervals called pumping periods. Within a pumping period, the well must pump at a constant rate. Any period of time up to 2 years is allowable as a pumping period.

10. A well may occupy any node position of the grid imposed on the plan view of the aquifer. The node point represents an area. The response coefficient calculated for a node may represent the effect of more than one well. The number of wells at a node does not vary over the design horizon.

11. The natural recharge and discharge to and from the aquifer are not disrupted by the pumping from wells.

12. Wells may be assumed to fully penetrate the aquifer.

Theory

The satisfying of the above conditions is sufficient to insure that the aquifer's response to pumping stress can be modeled with a linear ground water model. It is therefore possible to construct response coefficients that relate pumping at wells to the drawdown in those wells (Maddock, 1972). The coefficients exist even if the aquifer has irregularly shaped boundaries or nonhomogeneous flow parameters, such as transmissivity and storage coefficient.

The coefficients for an aquifer penetrated by M wells is derived as follows. When the previous stated conditions hold, flow within the aquifer is modeled by the partial differential equation (Maddock, 1972):

$$\nabla \cdot [T(\hat{x})\ \nabla s(\hat{x},t)] = S(\hat{x})\frac{\partial s}{\partial t}(\hat{x},t) - \sum_{j=1}^{M} Q'(\hat{x}_j,t)\delta(\hat{x}-\hat{x}_j) \qquad (1)$$

where $\hat{x}$ is the point (x,y) of observation, $T(\hat{x})$ is nonhomogeneous transmissivity, $S(\hat{x})$ is nonhomogeneous coefficient of storage for a confined aquifer or specific yield for an unconfined aquifer, $s(\hat{x},t)$ is the drawdown at point $\hat{x}$ at time t, $Q'(\hat{x}_k,t)$ is the instantaneous discharge at the k^{th} well at time t, and $\delta(\hat{x}-\hat{x}_j)$ is a Dirac delta function. Assuming no previous development, the initial condition on drawdown is

$$s(\hat{x},o) = 0 \qquad \hat{x} \text{ within and on boundary} \qquad (2)$$

If there are not rivers or streams present and if the pumping does not interfere with the natural recharge and discharge from the aquifer, the boundary conditions are

$$\frac{\partial s}{\partial n}(\lambda) = 0 \qquad (3)$$

where λ is a parameter indicating that $\partial s/\partial n$ is evaluated on the boundary. The boundary is irregular in shape, n is the normal direction and $\partial s/\partial n(\lambda)$ is the gradient of the drawdown for the normal to the boundary. If there are constant head conditions on portions of the boundary λ', the boundary conditions on that portion are

$$s(\lambda') = 0 \qquad (4)$$

Equation (1) and its initial and boundary conditions are linear, hence there exists a Green's function such that

$$s(\hat{x},t) = \int_{\forall \hat{x}} \int_{0}^{t} G(\hat{x},\hat{x}',t-\tau)F(\hat{x}',\tau)\, d\hat{x}'d\tau \tag{5}$$

where

$$F(\hat{x}',\tau) = \sum_{j=1}^{M} Q'(\hat{x}_j,\tau)\delta(\hat{x}'-\hat{x}_j) \tag{6}$$

Let the design period consist of N equal duration time periods of length η. When $Q'(\hat{x}_j,\tau)$, $j=1,\ldots,M$, are constants between the i^{th} and $i-1^{th}$ time periods, $i=1,\ldots,N$ the drawdown at the k^{th} well at the end of the n^{th} time period, rewritten $s(k,n)$, is given by the equation

$$s(k,n) = \sum_{j=1}^{M} \sum_{i=1}^{n} Q(j,i) \int_{(i-1)\eta}^{i\eta} G(\hat{x}_k,\hat{x}_j,n\,\eta-\tau)d\tau \tag{7}$$

where

$$Q(j,i) = Q'(\hat{x}_j,i\eta)\eta \tag{8}$$

is the quantity of water withdrawn from the j^{th} well in the i^{th} time period. Define

$$\beta(k,j,n-i+1) = \int_{(i-1)\eta}^{i\eta} G(\hat{x}_k,\hat{x}_j,n\,\eta-\tau)d\tau \tag{9}$$

and equation (7) becomes

$$s(k,n) = \sum_{j=1}^{M} \sum_{i=1}^{n} Q(j,i)\beta(k,j,n-i+1) \tag{10}$$

The β's are the response coefficients and are constants independent of pumping and drawdown. $\beta(k,j,n-i+1)$ measures the increment of drawdown at the k^{th} well at the n^{th} time period due to pumping at the j^{th} well during the i^{th} time period. The coefficients β's are related to the well hydraulics, being functions of the distances between wells, the well radii (grid size), the transmissivity of the aquifer, the storage coefficient when the aquifer is confined and the specific yield when the aquifer is unconfined, the boundary conditions, the initial conditions, and the type of partial differential equation chosen to emulate the flow phenomena. In practice the β's are determined by a simulation model because the irregularly shaped boundaries and nonhomogenenous parameters make analytical determination impossible. Maddock (1972) does determine a set of β's for an aquifer of infinite extent and homogeneous parameter by analytical methods. Equation 9 forms a linear algebraic technological function (LATF).

The Green's function $G(\hat{x},\hat{x}',t)$ is determined by solving the equation

$$\nabla \cdot [T(\hat{x})\nabla G(\hat{x},\hat{x}',t)] = S(\hat{x})\frac{\partial G}{\partial t}(\hat{x},\hat{x}',t) - \delta(\hat{x},x')\delta(t) \qquad (11)$$

numerically subject to the causality condition

$$G(\hat{x},\hat{x}', t-\tau) = 0 \quad t < \tau \qquad (12)$$

and the boundary conditions

$$\frac{\partial G(\hat{\lambda},t)}{\partial \eta} = 0 \text{ and/or } G(\lambda',t) = 0 \qquad (13)$$

Applying finite difference techniques to equation (11) leads to a discrete form of the equation

$$\frac{T_{i-\frac{1}{2}j}}{\Delta x^2}(G_{i-1jk} - G_{ijk}) + \frac{T_{i+\frac{1}{2}j}}{\Delta x^2}(G_{i+1jk} - G_{ijk})$$

$$+ \frac{T_{ij-\frac{1}{2}}}{\Delta x^2}(G_{ij-1k} - G_{ijk}) + \frac{T_{ij+\frac{1}{2}}}{\Delta x^2}(G_{ij+1k} - G_{ijk}) - \frac{S}{\Delta t} G_{ijk}$$

$$= - \begin{cases} \frac{S}{\Delta t} G_{ijk-1} + 1 & \text{if } k = 1 \\ \frac{S}{\Delta t} G_{ijk} - 1 & \text{if } k > 1 \end{cases} \tag{14}$$

A square grid has been used. The index k represents the k^{th} time step and i and j represent row and column number respectively of the square grid imposed to produce the discretized form. The i index represents the discrete form of y and j represents the discrete form of x. The half node expansions of transmissibility variable are defined as (Saul'yev, 1964).

$$T_{i-\frac{1}{2}j} = \frac{T_{i-1j} + T_{ij}}{2} \tag{15}$$

$$T_{i+\frac{1}{2}j} = \frac{T_{i+1j} + T_{ij}}{2} \tag{16}$$

$$T_{ij-\frac{1}{2}} = \frac{T_{ij-1} + T_{ij}}{2} \tag{17}$$

$$T_{ij+\frac{1}{2}} = \frac{T_{ij+1} + T_{ij}}{2} \tag{18}$$

An alternating-direction-implicit method is used to solve equation (14). In addition an iterative procedure is used to increase the speed and accuracy of the method (Rubin, 1968). Application of these techniques to equation (14) leads to two new equations

$$T_{ij-\frac{1}{2}}\, G^{n+\frac{1}{2}}_{ij-1k+1} - [T_{ij-\frac{1}{2}} + T_{ij+\frac{1}{2}} + \rho + I^{n+\frac{1}{2}}_{ij}]G^{n+\frac{1}{2}}_{ijk+1} + T_{y+\frac{1}{2}}\, G^{n+\frac{1}{2}}_{ijk+1}$$

$$= -T_{i-\frac{1}{2}j}\, G^{n}_{i-1jk+1} - \rho\, G^{n}_{ijk} + [T_{i-\frac{1}{2}j} + T_{i+\frac{1}{2}j} - I^{n+\frac{1}{2}}_{ij}]\, G^{n}_{ijk+1}$$

$$- T_{i+\frac{1}{2}j}\, G^{n}_{i+1jk+1} + \Delta x^{2}\delta(k-1) \tag{19}$$

and

$$T_{i-\frac{1}{2}j}\, G^{n+1}_{i-1jk+1} - [T_{i-\frac{1}{2}j} + T_{i+\frac{1}{2}j} + \rho + I^{n+1}_{ij}]G^{n+1}_{ijk+1} + T_{i+\frac{1}{2}j}\, G^{n+1}_{i+1jk+1}$$

$$= -T_{ij-\frac{1}{2}}\, G^{n+\frac{1}{2}}_{ij-1k+1} - \rho G^{n+\frac{1}{2}}_{ijk} + [T_{ij-\frac{1}{2}} + T_{ij+\frac{1}{2}} - I^{n+1}_{ij}]\, G^{n+\frac{1}{2}}_{ijk+1}$$

$$- T_{ij+\frac{1}{2}}\, G^{n+\frac{1}{2}}_{ij+1k+1} + \Delta x^{2}\delta(k-1) \tag{20}$$

where n is the iteration index,

$$\rho = \frac{\Delta x^{2} S}{\Delta t}\ , \tag{21}$$

I^{n}_{ij} denotes the iteration parameter at the (i,j) node and is defined

$$I^{n}_{ij} = I^{n-1}_{ij}\ \exp\left\{\frac{\ln \frac{2x_m}{\lambda-1}}{\lambda - 1}\right\} \tag{22}$$

with

λ -- the number of iteration parameters desired,

x_m -- the larger of the total number of rows or total of columns for the grid

initially

$$I^{1}_{ij} = \frac{\pi^{2}}{2x_m}\ [T_{i-\frac{1}{2}j} + T_{i+\frac{1}{2}j} + T_{ij-\frac{1}{2}} + T_{ij+\frac{1}{2}}]\ , \tag{23}$$

and

$$\delta(k-1) = \begin{cases} 1 & k = 1 \\ 0 & k \neq 1 \end{cases} \tag{24}$$

Equations (19) and (20) form two systems of linear equations. The first set represents the row values with column values held fixed, and the second set represents the column values with row values held fixed. The values of G^{n}_{ijk+1} and G^{n+1}_{ijk+1} are compared for all i's and j's. If the absolute value of their difference is less than some convergence factor, ε, the time increment counter, k, is advanced to k + 1, if not, the iteration parameter index, n, is advanced and the k index remains fixed. The row and column solutions are repeated for fixed index k, but advancing iteration parameter index until convergence is achieved. The parameters are cycled if more than the λ iterations are required. The time duration, Δt, represented by the index advance from k to k + 1, may be quite large. Values of up to two years have been used for Δt. As Δt increases more iterations with the iteration parameter are needed to achieve convergence.

Preliminary Operation

Values for transmissivity are given at each node of a grid superimposed on the plan view of the aquifer. The element of lengths x and y are equal. The nodal array of the grid is pictured as follows:

	Δx		
	i-1,j-1	i-1,j	i-1,j+1
Δy	i,j-1	i,j	i,j+1
	i+1,j-1	i+1,j	i+1,j+1

The rules for assigning the values of the hydrologic variables and parameters are:

1. If a node point lies outside the boundary of aquifer, the transmissivity value for that node is set equal to zero.
2. Values of transmissivity within the boundaries of the aquifer should have values whose dimensions are in square feet per second. The dimensions of the matrix of the transmissivity values, zero or non-zero may not exceed 50 by 50 unless the dimension statements in the program are changed.
3. Storage coefficient is a dimensionless constant everywhere including those areas outside the boundaries of aquifer.
4. Those node points that have a constant head value are flagged. (See section of Constant Head Boundary cards.)
5. Those node points that have wells are flagged.

When all the input data have been amassed, formatted, and punched onto cards as outlined in the Input Requirements and Data Description section, the program is ready to go.

Structure

The program consists of a main program and seven subroutines. The functions of the subroutines are as follows:

1. INF provides information as to type of boundary, type of output desired, initial values of variables, and the transmissivity values for the aquifer. These variables are uneffected by locations of wells.

2.	PARAM	provides calculations of the iteration parameters for the implicit iterative method
3.	BOUND	reads location signals for constant head boundary nodes and well nodes
4.	INF2	assigns the producing well - INF2 is called for each well node
5.	ITRATE	provides control over the iteration procedures
6.	MATCAL	MATCAL is three subroutines; INITL, ROW, and COLUMN and they are entered by multiple entry procedures. The three subroutines initialize the variable used in the solution technique (INITL), calculate a solution with column values fixed, (ROW) and then calculate a solution with the row values fixed (COLUMN).
7.	BETA	calculates the response coefficients

Input Requirements and Data Description

Input to the program consists of control cards and data cards

Control Cards

There are two control cards read at the beginning of the program: the first is called a suppression card and the second is called a format card.

Suppression Card.-- This card indicates whether or not constant head boundaries are present, punched output is required, write out on a file is required, or normal printout is desired. The suppression card variables are all integers

Column	Variable Name	
1 - 2	IBOUND	Constant head boundary signal: If IBOUND=1, then there exists somewhere in the aquifer a constant head boundary and that constant

Column	Variable Name	
		Head boundary location cards to be read. If there is no constant head boundary present, leave the field of IBOUND blank.
3 - 4	IDISK	File write out signal: If IDISK=1 then the β's are written out and stored unformatted on a file with unit number 10. If no file storage is required leave the field IDISK blank.
5 - 6	IPUNCH	Punched output signal: If IPUNCH=1 the β's are punched out on cards. The format for the punched output is provided by FBETA. If no punched output is required leave the field IPUNCH blank.
7 - 8	IWRITE	Printer output signal: If IWRITE=1 the β's are printed on the online printed. If no printout is required leave the field IWRITE blank.

Format Cards -- The format cards provide the format for reading in the transmissivity values and for punching out the β values (if required). This card is read at the time the program is executed so that the format of the data is dynamic. The card contains two 16 character fields. For a detailed explanation of preparing format cards see FORTRAN IV Language, an IBM publication (IBM C28-6515).

Column	Variable Name	
1 - 16	FTRAN	Transmissivity data format containing up to 16 characters.
17 - 32	FBETA	β punchout format of up to 16 characters. To avoid a syntax error a dummy format should be supplied even if no punched output is required.

Data Cards

There are two types of data cards: those that contain integer values and those that contain real values.

Integer Parameter Card. -- The integer parameter card provides the values for the grid dimension (both length and width), the number of well nodes, the number of time periods in the design horizon, and the number of iteration parameters desired. Each value must be right-justified.

Column	Variable Name	
1 - 8	IDIM	the total number of rows
9 - 16	JDIM	the total number of columns
17 - 24	NWEL	the number of well nodes
25 - 32	NUMKIT	the number of time periods in the design horizon. Up to 50 time periods are allowed as the program is now dimensioned
33 - 40	LENG	the number of iteration parameters

Real Parameter Card -- The real parameter card provides the values for the distance between node points, the constant storage coefficient, a scaling parameter for transmissivity (in case the user wishes to read the transmissivities in as integers), the duration of a pumping period, and a convergence factor to determine if the β's have been calculated to sufficient accuracy. Each value must be right-justified unless the decimal point is punched.

Card #	Column	Variable Name	
1	1-20	DX	the distance in feet represented by consecutive grid nodes
1	21-40	S	the coefficient of storage (dimensionless)

Card #	Column	Variable Name	
1	41-60	SCALE2	Scales integer transmissivity values to feet squared per second. Set SCALE2=1 if no scaling is required
1	61-80	DT	the duration of the pumping period in seconds
2	1 -20	EPS	If two consecutive iterations produce head values with a difference less than EPS, β's are calculated and the program proceeds to the next time step. EPS is thus a convergence factor.

Transmissivity Cards -- Transmissivity data should be read in a row at a time. This somewhat restricts the allowable format statements for FTRAN. The row number occupies the first field of the first card of a row. Transmissivity values for the row are then read from the remaining fields of the card. Cards will continue to be read until all the transmissivity values for the rows are in. Then a new row of transmissivity values along with its row number are read. The process continues until all the rows have been entered. The format for the transmissivity cards is specified in FTRAN. An example is (20I4), twenty 4-digit fields. The last card of the set of transmissivity cards is a check card. In columns 2 through 25 of the checkcard the words

TRANSMISSIVITY READ IN

should be punched. This card is read and is printed out. If the above words appear in the output, the transmissivity values probably have read in properly.

Constant Head Boundary Cards -- Those node points which lie on the constant head boundary are read one node point per card. If no constant head boundary is present (IBOUND=0 or blank on the suppression card), no node points are read. If IBOUND=1 there is a constant head boundary. Each of the cards provides the row and column number of the node and a signal flag which determines if another constant boundary node card is to be read. Each value must be right justified.

Column	Variable Name	
1-4	I	The row number for the constant head node
5-8	J	The column number for the constant head node
9-12	JSIG	A signal flag. If JSIG=0 another node card is read. If JSIG=1 no more node cards are read.

Well Cards -- The well cards read in the row and column number of the well nodes. A node may be representative of more than one well. As the program is now dimensioned, up to 100 well nodes are allowed.

The row number is followed by a column number. The format for the card is 20I4 thus there are ten well nodes per card (10 row numbers and 10 column numbers). For example, if I(k) is the row number of k^{th} well and J(k) is the column number of the k^{th} well, then for 2 wells the card would be as follows:

Column	Variable	
1-4	I(1)	row number of first well
5-8	J(1)	column number of first well
9-12	I(2)	row number of second well
13-16	J(2)	column number of second well

Each value must be right-justified. Once the well cards are read in, the input data requirements have been completed.

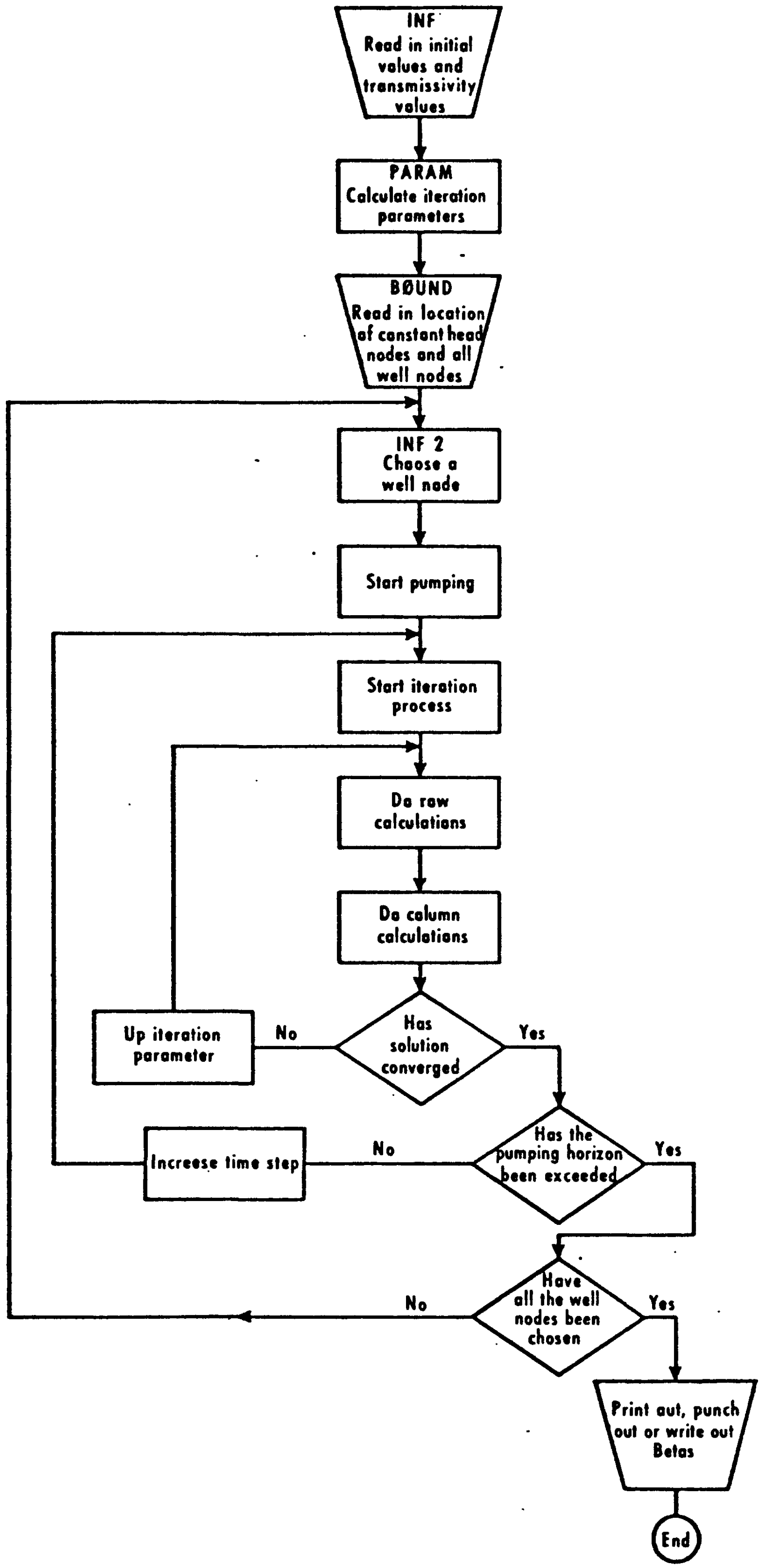
Output

The program output consists of 1) printing the integer and real variable inputs, 2) listing well nodes by row and column number, and 3) the option to print-out, punch-out, or write out (on a file) the response coefficient, β's.

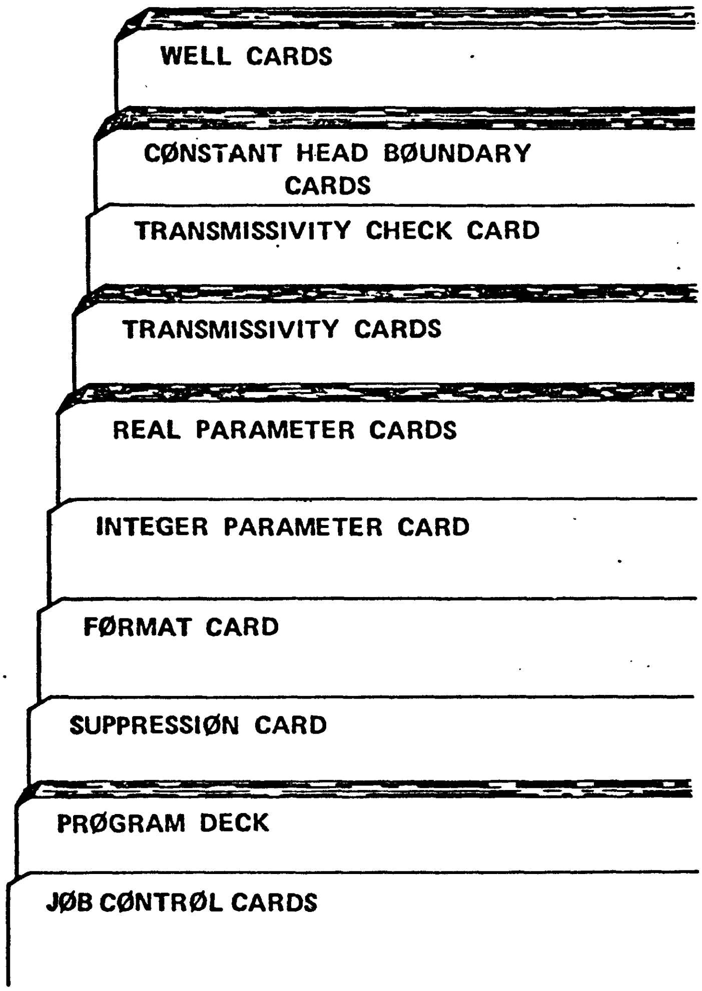
Core Requirements and Time Estimates

The program as it is now written and when it is run on an IBM 370-155 requires 248 K bytes of storage, 9 sec to compile, and averages 2 minutes per well node calculation (based on a design horizon of 20 one-year time periods). The calculating of the response coefficient for 30 well nodes over 20 one-year time periods takes about 60 minutes of compute time on an IBM 370-155. On an IBM 360-91 the same problem requires less than 10 minutes of computer time.

FLOW CHART


INF
Read in initial values and transmissivity values
PARAM
Calculate iteration parameters
BØUND
Read in location of constant head nodes and all well nodes
INF 2
Choose a well node
Start pumping
Start iteration process
Do row calculations
Do column calculations
Has solution converged
No
Up iteration parameter
Yes
Has the pumping horizon been exceeded
No
Increase time step
Yes
Have all the well nodes been chosen
No
Yes
Print out, punch out or write out Betas
End

LOADING PROCEDURE



References

Maddock, T, III, 1972, Algebraic technological function from a simulation model: Water Resources Research, vol. 8, no. 1, p. 129,134.

Rubin, J., 1968, Theoretical analysis of two dimensional transient flow of water in unsaturated and partly unsaturated soils: Proc. Soil Sci. Soc. Am., vol. 32, no. 5, p. 607-615.

Saul'yev, V.K., 1964, Integration of equations of parabolic type by the methods of nets: Pergamon Press, Oxford, England, 364 p.

APPENDIX A

Input Data to Program

```
   0   0   1   1
(20I4)          (2I3/(10F8.4))
      49      41       3      10      10
2640.0          0.01            1000.0          31557600.0
0.001
   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   3   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  25  25
  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   4   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  25  25
  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   5   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  25  25
  25  25  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   6   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  25  25
  25  25  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   7   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  25  25
  25  25  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   8   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  25  25
  25  25  25  25  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
   9   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  38  38
  38  38  25  25  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
  10   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  25  38  38
  38  38  25  25  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0
  11   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  38  38  38
  38  38  38  25  25  25  25  25  25  25  25  25  25  25  25   0   0   0   0   0
   0   0
  12   0   0   0   0   0   0   0   0   0   0   0   0   0  25  25  25  38  38  38
  38  38  38  25  25  25  25  25  25  25  25  25  25  25  25   0   0   0   0   0
   0   0
  13   0   0   0   0   0  25  25  25  25  25  25  25  25  25  25  38  38  38  38
  38  38  38  25  25  25  25  25  25  25  25  25  25  25  25   0   0   0   0   0
   0   0
  14   0   0   0   0   0  25  25  25  25  25  25  25  25  25  38  38  38  38  38
  38  38  38  25  25  25  25  25  25  25  25  25  25  25  25   0   0   0   0   0
   0   0
  15   0   0  25  25  25  25  25  25  25  25  25  25  25  38  38  38  38  38  38
  38  38  38  38  25  25  25  25  25  25  25  25  25  25  25   0   0   0   0   0
   0   0
  16   0   0   0  25  25  25  25  25  25  38  38  38  38  38  38  38  38  38  38
  38  38  38  38  38  25  25  25  25  25  25  25  25   0   0   0   0   0   0   0
   0   0
  17   0   0   0   0  25  25  25  38  38  38  38  38  38  38  38  38  38  38  38
  38  38  38  38  38  38  38  25  25  25  25  25  25  25  25   0   0   0   0   0
   0   0
  18   0   0   0   0   0   0  38  38  38  38  38  38  38  38  38  38  38  38  38
  38  38  38  38  38  38  38  38  38  38  25  25  25  25  25   0   0   0   0   0
   0   0
  19   0   0   0   0   0   0   0  38  38  38  38  38  38  38  38  38  38  38  38
  38  38  38  38  38  38  38  38  38  38  38  25  25  25  25   0   0   0   0   0
```

0	0																		
20	0	0	0	0	0	0	0	0	38	38	38	38	38	38	38	38	38	38	38
38	38	38	38	38	38	38	38	38	38	38	25	25	25	25	0	0	0	0	0
0	0																		
21	0	0	0	0	0	0	0	0	0	38	38	38	38	38	38	38	38	38	38
38	38	38	38	38	38	38	38	38	38	38	25	25	25	25	0	0	0	0	0
0	0																		
22	0	0	0	0	0	0	0	0	0	0	38	38	38	38	38	38	38	38	38
38	38	38	38	38	38	38	38	38	38	38	25	25	25	25	0	0	0	0	0
0	0																		
23	0	0	0	0	0	0	0	0	0	0	0	77	77	77	77	77	77	77	77
38	38	38	38	38	38	38	38	38	38	38	25	25	25	25	0	0	0	0	0
0	0																		
24	0	0	0	0	0	0	0	0	0	0	0	0	0	77	77	77	77	77	77
77	77	38	38	38	38	38	38	38	38	38	25	25	25	25	0	0	0	0	0
0	0																		
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	77	77	77	77
77	77	77	38	38	38	38	38	38	38	38	38	25	25	25	0	0	0	0	0
0	0																		
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	77	77	77	77
77	77	77	77	38	38	38	38	38	38	38	38	25	25	25	0	0	0	0	0
0	0																		
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	77	77	77	77
77	77	77	77	77	38	38	38	38	38	38	38	25	25	25	0	0	0	0	0
0	0																		
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	77	77	77
77	77	77	77	77	38	38	38	38	38	38	38	25	25	25	0	0	0	0	0
0	0																		
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	154	77	77
77	77	77	77	77	38	38	38	38	38	38	38	25	25	25	25	25	0	0	0
0	0																		
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	154	77
77	77	77	77	77	77	38	38	38	38	38	38	25	25	25	25	25	25	0	0
0	0																		
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	154	77
77	77	77	77	77	77	38	38	38	38	38	38	25	25	25	25	25	25	25	0
0	0																		
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	77
77	77	77	77	77	77	38	38	38	38	38	38	25	25	25	25	25	25	25	0
0	0																		
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	77
77	77	77	77	77	77	38	38	38	38	38	25	25	25	25	25	25	25	25	0
0	0																		
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77
77	77	77	77	77	77	38	38	38	38	25	25	25	25	25	25	25	25	25	0
0	0																		
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77
77	77	77	77	77	38	38	38	38	38	25	25	25	25	25	25	25	25	25	25
0	0																		
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	77	77	77	77	38	38	38	38	38	25	25	25	25	25	25	25	25	25	25
0	0																		
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	77	77	77	77	38	38	38	38	25	25	25	25	25	25	25	25	25	25	25
0	0																		
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	77	77	77	38	38	38	38	38	25	25	25	25	25	25	25	25	25	0	0
0	0																		
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	77	77	77	38	38	38	38	25	25	25	25	25	25	25	25	25	0	0	0
0	0																		

```
40   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0  77  77  77  38  38  38  38  25  25  25  25  25  25   0   0   0   0   0   0
 0   0
41   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0  77  77  38  38  38  38  25  25  25  25  25  25   0   0   0   0   0   0
 0   0
42   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0  77  77  38  38  38  38  25  25  25  25  25  25   0   0   0   0   0   0
 0   0
43   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0  77  77  38  38  38  38  25  25  25  25  25  25   0   0   0   0   0   0
 0   0
44   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0  38  38  38  38  38  38  25  25  25   0   0   0   0   0   0   0   0
 0   0
45   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0  38  38  38  38  38  38  25  25  25   0   0   0   0   0   0   0   0
 0   0
46   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0  38  38  38  38  38  25  25   0   0   0   0   0   0   0   0
 0   0
47   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0  25  25  25   0   0   0   0   0   0   0   0
 0   0
48   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0
49   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   0
TRANSMISSIVITY READ IN
 8  17   8  19  12  17
```

APPENDIX B

Program Listing

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C      ACIP-ITERATIVE
                C      BETA CALCULATIONS
                C      AQUIFER MODEL
                C      CONFINED AQUIFER OR UNCONFINED AQUIFER WITH CONSTANT
                C                            TRANSMISSIBILITY IN TIME
                C
                C      THIS PROGRAM USES AN IMPLICIT ALTERNATING DIRECTION SCHEME.
                C
  ISN 0002             COMMON /C8/ KT
  ISN 0003             COMMON /C10/ NUMKT,NWEL
  ISN 0004             COMMON /C23/ IDIVG,KOUNT
  ISN 0005             COMMON /C24/ LENG
                C
  ISN 0006             CALL INF
  ISN 0007             CALL PARAM
  ISN 0008             CALL BOUND
  ISN 0009             DO 3 KWEL=1,NWEL
  ISN 0010             CALL INF2(KWEL)
  ISN 0011             DO 2 KT=1,NUMKT
  ISN 0012           1 CALL ITRATE
  ISN 0013             CALL ROW
  ISN 0014             CALL COLUMN
  ISN 0015             IF (KOUNT.LE.5*LENG.AND.IDIVG.EQ.1) GO TO 1
  ISN 0017             IF (KOUNT.GT.5*LENG) WRITE (6,4)
  ISN 0019             IF (KOUNT.GT.5*LENG) GO TO 3
  ISN 0021             IF (IDIVG.EQ.0) CALL BETA(KWEL)
  ISN 0023           2 CONTINUE
  ISN 0024           3 CONTINUE
  ISN 0025             STOP
                C
                C
  ISN 0026           4 FORMAT (' THE SOLUTION FAILED TO CONVERGE')
  ISN 0027             END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE INF
                C
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                C
ISN 0004            COMMON /C1/ IDIM,JDIM
ISN 0005            COMMON /C2/ T(50,50)
ISN 0006            COMMON /C3/ H(50,50),STRTH
ISN 0007            COMMON /C4/ DT
ISN 0008            COMMON /C6/ PI
ISN 0009            COMMON /C9/ S
ISN 0010            COMMON /C10/ NUMKT,NWEL
ISN 0011            COMMON /C11/ IBOUND,IDISK,IPUNCH,IWRITE
ISN 0012            COMMON /C13/ DX
ISN 0013            COMMON /C18/ SCALE2
ISN 0014            COMMON /C19/ FTRAN(4),FBETA(4)
ISN 0015            COMMON /C21/ PARM,PREVH(50,50),EPS
ISN 0016            COMMON /C24/ LENG
                C
ISN 0017            INTEGER T,FTRAN,FBETA
                C
ISN 0018            NAMELIST /N1/ IBOUND,IDISK,IPUNCH,IWRITE/N2/LENG/N3/SCALE2,EPS
                C
                C   READ IN SIGNALS FOR SUPRESSION OR OPERATION OF SUBROUTINES
                C
ISN 0019            READ (5,7) IBOUND,IDISK,IPUNCH,IWRITE
ISN 0020            WRITE (6,N1)
                C
                C   READ IN FORMATS FOR TRANSMISSIVITY AND PUMPING CARDS.
                C
ISN 0021            READ (5,6) FTRAN,FBETA
ISN 0022            WRITE (6,6) FTRAN,FBETA
                C
                C   READ IN INTEGER VARIABLES
                C
ISN 0023            READ (5,3) IDIM,JDIM,NWEL,NUMKT,LENG
ISN 0024            WRITE (6,N2)
                C
                C   READ IN REAL VARIABLES
                C
ISN 0025            STRTH=1000.
ISN 0026            READ (5,4) DX,S,SCALE2,DT,EPS
ISN 0027            WRITE (6,N3)
                C
ISN 0028            PI=3.14159265359
ISN 0029            CALL INITL
                C
                C   READ IN TRANSMISSIBILITY MATRIX
                C
ISN 0030            DO 1 I=1,IDIM
ISN 0031          1 READ (5,FTRAN) I,(T(I,J),J=1,JDIM)
```

```
ISN 0032          READ (5,2)                                                          B 510
ISN C033          WRITE (6,2)                                                         B 520
            C                                                                         B 530
ISN 0034          WRITE (6,5) DT,DX,S,NUMKT,IDIM,JDIM,NWEL                            B 540
            C                                                                         B 550
ISN 0035          RETURN                                                              B 560
            C                                                                         B 570
            C                                                                         B 580
            C                                                                         B 590
ISN 0036        2 FORMAT ('                                                 ')        B 600
ISN C037        3 FORMAT (10I8)                                                       B 610
ISN C038        4 FORMAT (4F20.5)                                                     B 620
ISN 0039        5 FORMAT (1H1,1X,29HLENGTH OF INITIAL TIME STEP =,D16.4//1X,27HGRID   B 630
                 1SPACING OF PROTOTYPE =,F16.4//1X,21HSTORAGE COEFFICIENT =,F16.8//1  B 640
                 2X,30HMAXIMUM NUMBER OF TIME STEPS =,I4//1X,24HNUMBER OF LENGTH NOD  B 650
                 3ES =,I4//1X,23HNUMBER OF WIDTH NODES =,I4//1X,'NUMBER OF WELLS =',  B 660
                 4I4/)                                                                B 670
ISN 0040        6 FORMAT (2(4A4))                                                     B 680
ISN C041        7 FORMAT (10I2)                                                       B 690
ISN 0042          END                                                                 B 700-
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE PARAM
                  C
                  C     THIS SUBROUTINE COMPUTES ITERATION PARAMETERS
                  C
  ISN 0003              IMPLICIT REAL*8(A-H,O-Z)
                  C
  ISN 0004              COMMON /C1/ IDIM,JDIM
  ISN 0005              COMMON /C6/ PI
  ISN 0006              COMMON /C22/ OMEGA(10)
  ISN 0007              COMMON /C24/ LENG
                  C
  ISN 0008              XMAX=AMAX0(IDIM,JDIM)
  ISN 0009              OMIN=PI**2/(2.0*XMAX)
  ISN 0010              BETA=DEXP(DLOG(1.0/OMIN)/(DFLOAT(LENG)-1.0))
  ISN 0011              OMEGA(1)=OMIN
  ISN 0012              DO 1 I=2,LENG
  ISN 0013            1 OMEGA(I)=OMEGA(I-1)*BETA
  ISN 0014              RETURN
                  C
  ISN 0015              END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=C000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002           SUBROUTINE BOUND
              C
              C    THIS SUBROUTINE READS IN POINTS OF CONSTANT HEAD AND READS IN
              C    WELL NODES
              C
ISN 0003           IMPLICIT REAL*8(A-H,O-Z)
              C
ISN 0004           COMMON /C1/ IDIM,JDIM
ISN 0005           COMMON /C5/ P(50,50)
ISN 0006           COMMON /C10/ NUMKT,NWEL
ISN 0007           COMMON /C11/ IBOUND,IDISK,IPUNCH,IWRITE
ISN 0008           COMMON /C14/ II(70),JJ(70)
              C
ISN 0009           INTEGER P,FTRAN,FPUMP
              C
ISN 0010           DO 1 I=1,IDIM
ISN 0011           DO 1 J=1,JDIM
ISN 0012         1 P(I,J)=0
              C
              C    READ IN CONSTANT HEAD POINTS IF ANY
              C
ISN 0013           IF (IBOUND.EQ.0) GO TO 3
ISN 0015         2 READ (5,5) I,J,JSIG
ISN 0016           P(I,J)=2
ISN 0017           IF (JSIG.EQ.0) GO TO 2
ISN 0019         3 CONTINUE
              C
              C    READ IN WELL POINTS
              C
ISN 0020           READ (5,5) (II(K),JJ(K),K=1,NWEL)
ISN 0021           WRITE (6,4) (K,II(K),JJ(K),K=1,NWEL)
ISN 0022           RETURN
              C
              C
ISN 0023         4 FORMAT (1X,'WELL POINTS',5X,'I',4X,'J'/(5X,I2,9X,I2,3X,I2))
ISN 0024         5 FORMAT (20I4)
ISN 0025           END
```

```
LEVEL 21.7 ( JAN 73 )                    OS/360  FORTRAN H                         DATE  74.345/08.22.37

       COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0)(OK,
                              SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              SUBROUTINE INF2(KWEL)                                          E  10
                C                                                                      E  20
  ISN 0003              IMPLICIT REAL*8(A-H,O-Z)                                       E  30
                C                                                                      E  40
  ISN 0004              COMMON /C1/ IDIM,JDIM                                          E  50
  ISN 0005              COMMON /C3/ H(50,50),STRTH                                     E  60
  ISN 0006              COMMON /C5/ P(50,50)                                           E  70
  ISN 0007              COMMON /C11/ IBOUND,IDISK,IPUNCH,IWRITE                        E  80
  ISN 0008              COMMON /C14/ II(70),JJ(70)                                     E  90
  ISN 0009              COMMON /C23/ IDIVG,KOUNT                                       E 100
                C                                                                      E 110
  ISN 0010              INTEGER P                                                      E 120
                C                                                                      E 130
                C       INITIALIZE CONSTANTS                                           E 140
                C                                                                      E 150
  ISN 0011              IDIVG=0                                                        E 160
                C                                                                      E 170
                C       SET INITIAL VALUE OF HEAD AND ZERO PUMPING                     E 180
                C                                                                      E 190
  ISN 0012              DO 1 I=1,IDIM                                                  E 200
  ISN 0013              DO 1 J=1,JDIM                                                  E 210
  ISN 0014              IF (P(I,J).NE.2) P(I,J)=0                                      E 220
  ISN 0016            1 H(I,J)=STRTH                                                   E 230
                C                                                                      E 240
                C       SET UP PUMPING WELL                                            E 250
                C                                                                      E 260
  ISN 0017              I=II(KWEL)                                                     E 270
  ISN 0018              J=JJ(KWEL)                                                     E 280
  ISN 0019              P(I,J)=1                                                       E 290
  ISN 0020              IF (IWRITE.EQ.1) WRITE (6,2)                                   E 300
  ISN 0022              RETURN                                                         E 310
                C                                                                      E 320
                C                                                                      E 330
                C                                                                      E 340
  ISN 0023            2 FORMAT (1H1)                                                   E 350
  ISN 0024              END                                                            E 360-
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0000K,
                          SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002           SUBROUTINE ITRATE
               C
 ISN 0003           IMPLICIT REAL*8(A-H,O-Z)
               C
 ISN 0004           COMMON /C1/ IDIM,JDIM
 ISN 0005           COMMON /C3/ H(50,50),STRTH
 ISN 0006           COMMON /C4/ DT
 ISN 0007           COMMON /C10/ NUMKT,NWEL
 ISN 0008           COMMON /C21/ PARM,PREVH(50,50),EPS
 ISN 0009           COMMON /C22/ OMEGA(10)
 ISN 0010           COMMON /C23/ IDIVG,KOUNT
 ISN 0011           COMMON /C24/ LENG
               C
 ISN 0012           IF (IDIVG.EQ.1) GO TO 2
 ISN 0014           NTH=0
 ISN 0015           KOUNT=0
 ISN 0016           PARM=0.0
 ISN 0017           DO 1 I=1,IDIM
 ISN 0018           DO 1 J=1,JDIM
 ISN 0019         1 PREVH(I,J)=H(I,J)
 ISN 0020           RETURN
 ISN 0021         2 KOUNT=KOUNT+1
 ISN 0022           IF (MOD(KOUNT,LENG)) 3,3,4
 ISN 0023         3 NTH=0
 ISN 0024         4 NTH=NTH+1
 ISN 0025           PARM=OMEGA(NTH)
 ISN 0026           RETURN
 ISN 0027           END
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE MATCAL                                          G  10
              C                                                                G  20
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)                                   G  30
              C                                                                G  40
ISN 0004            COMMON /C1/ IDIM,JDIM                                      G  50
ISN 0005            COMMON /C2/ T(50,50)                                       G  60
ISN 0006            COMMON /C3/ H(50,50),STRTH                                 G  70
ISN 0007            COMMON /C4/ DT                                             G  80
ISN 0008            COMMON /C5/ P(50,50)                                       G  90
ISN 0009            COMMON /C6/ KT                                             G 100
ISN 0010            COMMON /C9/ S                                              G 110
ISN 0011            COMMON /C13/ DX                                            G 120
ISN 0012            COMMON /C18/ SCALE2                                        G 130
ISN 0013            COMMON /C21/ PARM,PREVH(50,50),EPS                         G 140
ISN 0014            COMMON /C23/ IDIVG,KOUNT                                   G 150
              C                                                                G 160
ISN 0015            INTEGER T,P                                                G 170
ISN 0016            DIMENSION BE(50), G(50), TEMP(50)                          G 180
              C                                                                G 190
ISN 0017            ENTRY INITL                                                G 200
              C                                                                G 210
ISN 0018            DIV=2.0*SCALE2                                             G 220
ISN 0019            JLESS1=JDIM-1                                              G 230
ISN 0020            JLESS2=JDIM-2                                              G 240
ISN 0021            ILESS1=IDIM-1                                              G 250
ISN 0022            ILESS2=IDIM-2                                              G 260
ISN 0023            RETURN                                                     G 270
              C                                                                G 280
ISN 0024            ENTRY ROW                                                  G 290
              C                                                                G 300
ISN 0025            RHO=DX**2*S/DT                                             G 310
ISN 0026            BE(1)=0.0                                                  G 320
ISN 0027            G(1)=0.0                                                   G 330
ISN 0028            DO 9 I=2,IDIM                                              G 340
ISN 0029            DO 1 J=2,JLESS1                                            G 350
              C                                                                G 360
              C     DETERMINE WHETHER NODE IS OUTSIDE AQUIFER BOUNDARY         G 370
              C                                                                G 380
ISN 0030            IF (T(I,J).EQ.0) GO TO 1                                   G 390
              C                                                                G 400
              C     CALCULATE AVERAGE VALUES OF T BETWEEN ADJACENT NODES.      G 410
              C                                                                G 420
              C     NODE.   T1=LEFT,  T2=RIGHT,  T3=UPPER,  T4=LOWER           G 430
              C                                                                G 440
ISN 0032            T1=DFLOAT(T(I,J-1)+T(I,J))/DIV                             G 450
ISN 0033            T2=DFLOAT(T(I,J+1)+T(I,J))/DIV                             G 460
ISN 0034            T3=DFLOAT(T(I-1,J)+T(I,J))/DIV                             G 470
ISN 0035            T4=DFLOAT(T(I+1,J)+T(I,J))/DIV                             G 480
ISN 0036            IF (T(I,J-1).EQ.0) T1=0.0                                  G 490
ISN 0038            IF (T(I,J+1).EQ.0) T2=0.0                                  G 500
```

```
ISN 0040          IF (T(I-1,J).EQ.0) T3=0.0
ISN 0042          IF (T(I+1,J).EQ.0) T4=0.0
ISN 0044          HINCR=PARM*(T1+T2+T3+T4)
           C
           C      DEFINE BOUNDARY REQUIREMENTS OF CALCULATION PARAMETERS A,B,C,G,BE
           C      FOR ROWS OF MATRIX
           C
           C
           C      CALCULATE VALUES  FOR PARAMETERS A,B,C,D
           C
ISN 0045          Q=0.0
ISN 0046          ZK=0.0
ISN 0047          IF (P(I,J).EQ.1.AND.KT.EQ.1) Q=1.0
ISN 0049          IF (P(I,J).EQ.2) ZK=10.0D+50
ISN 0051          A=T1
ISN 0052          B=-T1-T2-RHO-ZK*DX**2-HINCR
ISN 0053          C=T2
ISN 0054          D=-T3*H(I-1,J)-RHO*PREVH(I,J)+(T3+T4-HINCR)*H(I,J)-T4*H(I+1,J)+Q-(
                 1ZK*DX**2)*(STRTH-H(I,J)/2.0)
ISN 0055          W=B-A*BE(J-1)
ISN 0056          BE(J)=C/W
ISN 0057          G(J)=(D-A*G(J-1))/W
ISN 0058        1 CONTINUE
           C
           C      CALCULATE HEAD VALUES FOR ROWS OF MATRIX AND PLACE THEM IN
           C      TEMPORARY LOCATION TEMP
           C
ISN 0059          IF (I.GT.2) GO TO 5
ISN 0061          DO 4 K=1,JLESS2
ISN 0062          N=JDIM-K
ISN 0063          IF (T(I,N)) 3,2,3
ISN 0064        2 TEMP(N)=STRTH
ISN 0065          GO TO 4
ISN 0066        3 TEMP(N)=G(N)-BE(N)*TEMP(N+1)
ISN 0067        4 CONTINUE
ISN 0068          GO TO 9
ISN 0069        5 DO 8 K=1,JLESS2
ISN 0070          N=JDIM-K
ISN 0071          H(I-1,N)=TEMP(N)
ISN 0072          IF (T(I,N)) 7,6,7
ISN 0073        6 TEMP(N)=STRTH
ISN 0074          GO TO 8
ISN 0075        7 TEMP(N)=G(N)-BE(N)*TEMP(N+1)
ISN 0076        8 CONTINUE
ISN 0077        9 CONTINUE
           C
ISN 0078          RETURN
           C
ISN 0079          ENTRY COLUMN
           C
ISN 0080          IDIVG=0
ISN 0081          DO 18 J=2,JDIM
```

```
ISN 0082          DO 10 I=2,ILESS1
            C
            C     DETERMINE WHETHER NODE IS OUTSIDE AQUIFER BOUNDARY
            C
ISN 0083          IF (T(I,J).EQ.0) GO TO 10
            C
            C     DEFINE BOUNDARY REQUIREMENTS OF CALCULATION PARAMETERS A,B,C,BE
            C     FOR COLUMN OF MATRIX
            C
            C     CALCULATE AVERAGE VALUE OF T BETWEEN ADJACENT NODES
            C
ISN 0085          T1=DFLOAT(T(I,J-1)+T(I,J))/DIV
ISN 0086          T2=DFLOAT(T(I,J+1)+T(I,J))/DIV
ISN 0087          T3=DFLOAT(T(I-1,J)+T(I,J))/DIV
ISN 0088          T4=DFLOAT(T(I+1,J)+T(I,J))/DIV
ISN 0089          IF (T(I,J-1).EQ.0) T1=0.0
ISN 0091          IF (T(I,J+1).EQ.0) T2=0.
ISN 0093          IF (T(I-1,J).EQ.0) T3=0.0
ISN 0095          IF (T(I+1,J).EQ.0) T4=0.0
ISN 0097          HINCH=PARM*(T1+T2+T3+T4)
            C
            C     CALCULATE VALUES FOR PARAMETERS A,B,C,D
            C
ISN 0098          Q=0.0
ISN 0099          ZK=0.0
ISN 0100          IF (P(I,J).EQ.1.AND.KT.EQ.1) Q=1.0
ISN 0102          IF (P(I,J).EQ.2) ZK=10.0D+50
ISN 0104          A=T3
ISN 0105          B=-T3-T4-RHO-ZK*DX**2-HINCH
ISN 0106          C=T4
ISN 0107          D=-T1*H(I,J-1)-RHO*PREVH(I,J)+(T1+T2-HINCR)*H(I,J)-T2*H(I,J+1)+Q-(
                 1ZK*DX**2)*(STRTH-H(I,J)/2.0)
ISN 0108          W=B-A*BE(I-1)
ISN 0109          BE(I)=C/W
ISN 0110          G(I)=(D-A*G(I-1))/W
ISN 0111       10 CONTINUE
            C
            C     CALCULATE HEAD VALUES FOR COLUMNS OF MATRIX AND PLACE IN TEMPORARY
            C     LOCATION TEMP
            C
ISN 0112          IF (J.GT.2) GO TO 14
ISN 0114          DO 13 K=1,ILESS2
ISN 0115          N=IDIM-K
ISN 0116          IF (T(N,J)) 12,11,12
ISN 0117       11 TEMP(N)=STRTH
ISN 0118          GO TO 13
ISN 0119       12 TEMP(N)=G(N)-BE(N)*TEMP(N+1)
ISN 0120          IF (DABS(TEMP(N)-H(N,J)).GT.EPS) IDIVG=1
ISN 0122       13 CONTINUE
ISN 0123          GO TO 18
ISN 0124       14 DO 17 K=1,ILESS2
ISN 0125          N=IDIM-K
```

```
ISN 0126            H(N,J-1)=TEMP(N)                                          G1550
ISN 0127            IF (T(N,J)) 16,15,16                                      G1560
ISN 0128         15 TEMP(N)=STRTH                                             G1570
ISN 0129            GC TO 17                                                  G1580
ISN 0130         16 TEMP(N)=G(N)-BE(N)*TEMP(N+1)                              G1590
ISN 0131            IF (DABS(TEMP(N)-H(N,J)).GT.EPS) IDIVG=1                  G1600
ISN 0133         17 CCNTINUE                                                  G1610
ISN 0134         18 CCNTINUE                                                  G1620
               C                                                              G1630
ISN 0135            RETURN                                                    G1640
               C                                                              G1650
ISN 0136            END                                                       G1660-
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE BETA(KWEL)                                        H  10
             C                                                                 H  20
             C    THIS SUBROUTINE CALCULATES BETAS                             H  30
             C                                                                 H  40
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)                                     H  50
             C                                                                 H  60
ISN 0004          COMMON /C1/ IDIM,JDIM                                        H  70
ISN 0005          COMMON /C3/ H(50,50),STRTH                                   H  80
ISN 0006          COMMON /C8/ KT                                               H  90
ISN 0007          COMMON /C10/ NUMKT,NWEL                                      H 100
ISN 0008          COMMON /C11/ IRCUND,IDISK,IPUNCH,IWRITE                      H 110
ISN 0009          COMMON /C14/ II(70),JJ(70)                                   H 120
ISN 0010          COMMON /C19/ FTRAN(4),FBETA(4)                               H 130
             C                                                                 H 140
ISN 0011          INTEGER FTRAN,FBETA                                          H 150
ISN 0012          REAL *4B(100,20)                                             H 160
             C                                                                 H 170
             C    CALCULATE BETAS                                              H 180
             C                                                                 H 190
ISN 0013          DO 1 KOUNT=1,NWEL                                            H 200
ISN 0014          I=II(KOUNT)                                                  H 210
ISN 0015          J=JJ(KOUNT)                                                  H 220
ISN 0016        1 B(KOUNT,KT)=STRTH-H(I,J)                                     H 230
             C                                                                 H 240
             C    WRITE OUT BETAS                                              H 250
             C                                                                 H 260
ISN 0017          IF (IDISK.EQ.1) WRITE (10) (B(KOUNT,KT),KOUNT=1,NWEL)        H 270
ISN 0019          IF (IPUNCH.EQ.1) WRITE (7,FBETA) KWEL,KT,(B(KOUNT,KT),KOUNT=1,NWEL  H 280
                 1)                                                            H 290
ISN 0021          IF (IWRITE.EQ.1) WRITE (6,2) KWEL,KT,(B(KOUNT,KT),KOUNT=1,NWEL)  H 300
ISN 0023          RETURN                                                       H 310
             C                                                                 H 320
             C                                                                 H 330
             C                                                                 H 340
ISN 0024        2 FORMAT (1X,'BETA(',I2,',J,',I2,')',4X,(15F7.3))              H 350
ISN 0025          END                                                          H 360-
```

APPENDIX C

Program Output from Printe

```
&N1
IHCLND=          0,IDISK=          0,IPUNCH=          1,IWRITE=          1
&ENC
(20I4)         (2I3/(10F8.4))
&N2
LENG=         10
&ENC
&N3
SCALE2=  1000.00000000000     ,EPS= 0.9999999999999995D-03
&ENC
TRANSMISSIVITY READ IN
```

```
LENGTH OF INITIAL TIME STEP =      0.3156D 08

GRID SPACING OF PROTOTYPE =      2640.0000

STORAGE COEFFICIENT =      0.01000000

MAXIMUM NUMBER OF TIME STEPS =  10

NUMBER OF LENGTH NODES =  49

NUMBER OF WIDTH NODES =  41

NUMBER OF WELLS =   3

WELL POINTS      I    J
     1           8   17
     2           8   19
     3          12   17
```

BETA(1,J, 1)	19.135	5.512	2.451
BETA(1,J, 2)	4.531	3.536	2.236
BETA(1,J, 3)	3.026	2.745	1.894
BETA(1,J, 4)	2.476	2.321	1.657
BETA(1,J, 5)	2.126	2.021	1.486
BETA(1,J, 6)	1.864	1.785	1.354
BETA(1,J, 7)	1.656	1.595	1.245
BETA(1,J, 8)	1.488	1.439	1.155
BETA(1,J, 9)	1.350	1.310	1.078
BETA(1,J,10)	1.237	1.203	1.012

BETA(2,J, 1)	5.510	17.943	2.166
BETA(2,J, 2)	3.535	4.139	2.082
BETA(2,J, 3)	2.745	2.765	1.610
BETA(2,J, 4)	2.321	2.270	1.601
BETA(2,J, 5)	2.020	1.959	1.443
BETA(2,J, 6)	1.785	1.727	1.317
BETA(2,J, 7)	1.594	1.544	1.214
BETA(2,J, 8)	1.438	1.396	1.127
BETA(2,J, 9)	1.310	1.274	1.054
BETA(2,J,10)	1.203	1.174	0.992

BETA(3,J, 1)	2.435	2.154	14.471
BETA(3,J, 2)	2.231	2.076	3.018
BETA(3,J, 3)	1.897	1.811	1.836
BETA(3,J, 4)	1.661	1.605	1.456
BETA(3,J, 5)	1.490	1.447	1.264
BETA(3,J, 6)	1.357	1.320	1.145
BETA(3,J, 7)	1.248	1.216	1.059
BETA(3,J, 8)	1.157	1.130	0.992
BETA(3,J, 9)	1.080	1.056	0.938
BETA(3,J,10)	1.014	0.994	0.892

APPENDIX D

Programs Punched Output

```
 1  1
19.1353  5.5124  2.4509
 1  2
 4.5312  3.5382  2.2356
 1  3
 3.0258  2.7447  1.8942
 1  4
 2.4755  2.3208  1.6570
 1  5
 2.1259  2.0206  1.4864
 1  6
 1.8536  1.7849  1.3535
 1  7
 1.6561  1.5945  1.2452
 1  8
 1.4879  1.4387  1.1545
 1  9
 1.3503  1.3101  1.0778
 1 10
 1.2366  1.2035  1.0125
 2  1
 5.5100 17.9428  2.1663
 2 . 2
 3.5386  4.1390  2.0821
 2  3
 2.7451  2.7653  1.8098
 2  4
 2.3210  2.2703  1.6015
 2  5
 2.0203  1.9592  1.4432
 2  6
 1.7646  1.7274  1.3173
 2  7
 1.5944  1.5444  1.2140
 2  8
 1.4385  1.3962  1.1275
 2  9
 1.3100  1.2744  1.0545
 2 10
 1.2034  1.1736  0.9923
 3  1
 2.4349  2.1543 14.4712
 3  2
 2.2305  2.0764  3.0180
 3  3
 1.8965  1.8113  1.8362
 3  4
 1.6510  1.6046  1.4556
 3  5
 1.4903  1.4467  1.2643
 3  6
 1.3570  1.3203  1.1447
 3  7
 1.2481  1.2165  1.0590
 3  8
 1.1569  1.1296  0.9923
 3  9
 1.0797  1.0562  0.9377
 3 10
 1.0140  0.9937  0.8919
```